Master in Artificial Intelligence (UPC-URV-UB)

Master of Science Thesis

A Software System for the Microbial Source Tracking Problem

David Sanchez-Mendoza
david.sanchez-mendoza@est.fib.upc.edu

Advisor: *Lluís Belanche Muñoz*

June 20, 2012

L'univers és infinit,
pertot acaba i comença,
i ençà, enllà, amunt i avall,
la immensitat és oberta,
i a on tu veus lo desert
eixams de mons formiguegen.

Jacint Verdaguer, *Plus ultra*.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Lluís Belanche, for giving me the opportunity of working with him and for his wise advises (not necessarily related to academic stuff). I sincerely hope we can work together again in the future.

Besides my advisor, I would like to thank Anicet Blanch for providing the data we have been working with as well as his deep knowlegde in Microbial Source Tracking field.

I thank also my classmates just for having become friends and any people who have helped me in some way or another.

Last but not the least, I would like to thank my parents for always supporting me no matter what my decisions are.

Abstract

The *Microbial Source Tracking* problem (MST) has to do with the determination of the fecal pollution origin in waters by the use of microbial and chemical indicators. This document introduces a methodology for solving MST problem from the machine learning point of view reporting both the arising specific problems and challenges and how they have been addressed. The simplest instance of the MST problem has already been solved to satisfaction using machine learning techniques on recently and heavily polluted waters, however, our methodology accepts examples showing different concentration levels and using indicators (variables) with different environmental persistence. The theoretical methodology is supported by a software which implements it and has been validated using two real datasets with real data from different geographical and climatic areas.

Contents

1 Introduction **2**
- 1.1 Aims of the work . . . 3
- 1.2 Document organization . . . 3

2 Microbial source tracking **5**
- 2.1 Overview and relevance . . . 5
- 2.2 MST history . . . 6
- 2.3 Tracking methods . . . 6
  - 2.3.1 Library-dependent . . . 6
    - 2.3.1.1 Representativeness and proportionality . . . 7
    - 2.3.1.2 Geographic stability . . . 7
    - 2.3.1.3 Temporal stability . . . 7
    - 2.3.1.4 Summary . . . 7
  - 2.3.2 Library-independent . . . 8
    - 2.3.2.1 Basic requirements . . . 8
- 2.4 Modelling concentration level and environmental persistence . . . 9
  - 2.4.1 Concentration level . . . 9
  - 2.4.2 Time persistence . . . 9
  - 2.4.3 Detection threshold . . . 10
- 2.5 A real example . . . 11
  - 2.5.1 Heavily polluted water . . . 11
  - 2.5.2 Different concentration level and time persistence . . . 11

3 Learning methods and techniques **15**
- 3.1 Overview . . . 15
- 3.2 The learning problem . . . 16
  - 3.2.1 General formulation . . . 16
  - 3.2.2 Supervised learning . . . 18
  - 3.2.3 Unsupervised learning . . . 19
  - 3.2.4 Reinforcement learning . . . 19
- 3.3 Feature selection . . . 19
  - 3.3.1 Overview . . . 19
  - 3.3.2 Wrapper approach vs. filter approach . . . 20
  - 3.3.3 A feature selection algorithm example . . . 21
- 3.4 Validation protocol . . . 22
  - 3.4.1 Hold-out . . . 22
  - 3.4.2 Random sub-sampling . . . 23

3.4.3 K-fold cross-validation . . . 23
3.4.4 Leave-one-out cross-validation . . . 24
3.5 Prediction models . . . 24
3.5.1 Stabilised linear discriminant analysis . . . 24
3.5.2 Support vector machines . . . 25
3.5.3 K-nearest neighbor . . . 27

4 Related work **29**
4.1 Introduction . . . 29
4.2 Data modelling . . . 29
4.3 Experiments and results . . . 30
4.3.1 Experiment 1: 26 single variables, no feature selection . . . 30
4.3.2 Experiment 2: all variables, wrapper approach feature selection 30
4.3.3 Experiment 3: all variables except GA17, wrapper approach feature selection . . . 31
4.4 Conclusions . . . 31

5 Methodology **34**
5.1 Main strategy . . . 34
5.2 Dilution sections approach . . . 36
5.2.1 Building and selecting prediction models . . . 36
5.2.2 Dilution and ageing estimation . . . 37
5.2.3 New examples prediction . . . 38
5.3 Validation . . . 39

6 Application **41**
6.1 Real test dataset 1: the "Cyprus" data . . . 41
6.1.1 Data origin . . . 41
6.1.2 Data description . . . 42
6.1.3 Importance of variables regarding with prediction . . . 43
6.1.4 Importance of variables regarding with time and age estimation 47
6.1.5 Validation . . . 47
6.1.6 Test matrix prediction . . . 48
6.1.6.1 Diluted and aged data . . . 48
6.1.6.2 Point of source data . . . 48
6.1.6.3 Conclusions . . . 48
6.2 Real test dataset 2: the "Delta" data . . . 49
6.2.1 Data origin . . . 49
6.2.2 Data description . . . 50
6.2.3 Importance of variables regarding with prediction . . . 50
6.2.4 Validation . . . 53
6.2.5 Test matrix prediction . . . 54
6.2.5.1 "Llobregat" test matrix . . . 54
6.2.5.2 "Ebre" test matrix . . . 55

7 Conclusion and Future Work **57**
7.1 Conclusion 57
7.2 Future work: the no-dilution paradigm 57
7.2.1 Motivation 57
7.2.2 Strategy 58

List of Figures

2.1 Several measurements on FC indicator along the time. A linear regression on them is shown. 10
2.2 Distribution of samples according to indicators SOMCPH and SOMCPH/GA17 on heavily polluted water. 12
2.3 Distribution of samples according to indicators SOMCPH and SOMCPH/GA17 showing low concentration level. 13
2.4 Distribution of samples according to indicators SOMCPH and SOMCPH/GA17 showing low concentration level and high time persistence. 14

3.1 General learning scenario (extracted from [7]) 16
3.2 Support vector machine optimal hyperplane. 25

4.1 Percentages of correct classification provided by the different statistical methods tested for the development of predictive models using the 26 single variables defined in this study. TR: leave-one-out cross-validation performance in the training set (81 observations). TE: performance in the test set (22 observations). (Extracted from [5]) 30
4.2 Percentages of correct classification provided by the different methods tested for the development of predictive models using the lowest number of variables out of the 38 single and combined variables excluding the single and derived variables that use BTHPH. (Extracted from [5]) . . 32
4.3 Training observations according to the variables SOMCPH/BTHPH and SOMCPH. Variable values are standardized. Superimposed is the maximum-margin linear solution of the SVM. The three support vectors are indicated by a circle. (Extracted from [5]) 32

6.1 Distribution of samples of "Cyprus" matrix according to the geographical areas. (Extracted from [3]). 42
6.2 Indicators measured for "Cyprus" matrix.(Extracted from [3]). 43
6.3 Importance of variables from "Cyprus" matrix regarding with prediction. 44
6.4 SOMCPH/GA17 and FC/GA17 indicators. 45
6.5 GA17 and SOMCPH/GA17 indicators. 45
6.6 FRNAPH.I and CL indicators. 46
6.7 FRNAPH.III and FE inds. 46
6.8 GA17 and SOMCPH/GA17 indicators. 46
6.9 HBSA.Y/HBSA.T and FRNAPH.II indicators. 46
6.10 "Delta" data measurement place. 49

6.11 “Delta” data measurement place. 49
6.12 Importance of variables: cultivated indicators in summer. 51
6.13 Importance of variables: cultivated indicators in winter. 51
6.14 Good cultivated indicators: CW18 and SOMCPH/GA17. 51
6.15 Bad cultivated indicators: FE and SOMPCH. 51
6.16 Importance of variables: molecular indicators in summer. 52
6.17 Importance of variables: molecular indicators in winter. 52
6.18 Good molecular indicators: ADO, POMITO and CKMITO 53
6.19 Bad molecular indicators: DEN and BOMITO 53
6.20 Importance of variables: all indicators in summer. 54
6.21 Importance of variables: all indicators in winter. 54

List of Algorithms

1 Sequential Forward Selection 22
2 Estimating age 38
3 Estimating dilution degree 38
4 Building validation set 40

Chapter 1

Introduction

This document constitutes the final document of the master thesis belonging to the *Artificial Intelligence Master Program* organized by *Universitat Politècnica de Catalunya (UPC)*, *Universitat de Barcelona (UB)* and *Universitat Rovira i Virgili (URV)*.

Microbial source tracking (MST) is a recently coined term that includes different methodological approaches that pursuit the determination of the origin of fecal pollution in water by the use of microbial or chemical indicators [2].

Nowadays, fecal pollution in water is one of the main causes of health problems in the world, and is associated with several thousands of deaths per day, being a main vehicle of pathogen transmission. In this sense, is very important to know whether a waterbody (a river, a lake, etc.) is contaminated or not and, what is more, in case it is contaminated, which is the pollution origin.

The problem is important from both the scientific and legal points of view, for instance:

- In scientific terms key issues are accuracy and low complexity: we are particularly interested in those models that use a minimum number of variables, given the high technical and monetary costs that are implicit in the collection of this kind of data.
- On the other hand, contaminated water appears from time to time and the fact of knowing the contamination origin may help to determine who is responsible for it and, thus, who should be sanctioned or not.

There is a clear trend in MST studies to define specific indicators for different fecal sources and to establish standardized methodologies by an easy routine application for the enumeration of these indicators. Most of the research carried out has been focused on defining new indicators and suitable methodologies for detection and enumeration. Most of the many studies that develop predictive models for MST have been based on its definition at the *point of source*, assessing mainly the specificity and sensitivity of indicators and/or their combinations. Progressively, the need has arisen to assess the effects of the *dilution* of the contributions of fecal pollution in receiving waters as well as the *persistence* of these indicators on the environment.

1.1 Aims of the work

A simple instance of MST problem (differentiating human origin from non-human origin on recently and heavily polluted waters) has already been solved to satisfaction using machine learning techniques [5]. However, is always not possible to provide a data sample with the guarantee that it has been extracted from the *point of source*. In this sense it would be very useful to develop a methodology which is able to accept examples showing different concentration levels and using indicators (variables) with different environmental persistence, that is, examples that definitely have not been extracted from the *point of source*.

Precisely, our main contribution to the MST problem and the main aim of this research is designing a methodology which is able to deal and build predictive models not only with *point of source* data (recently and heavily polluted) but also with examples that show different concentration levels (affected by *dilution*) and different environmental persistence (affected by *time*) when differentiating human origin from non-human origin.

Apart from the methodology itself we have build a software system, named *Ichnaea*[1], which is written entirely in R [6] and implements our methodology allowing users to train the system with their own data and making their own predictions on it; this software also reports to the user which of the indicators have more discrimination power and, given a particular example to be predicted, proposes which ones of the not included indicators should be included in order to improve the prediction confidence.

1.2 Document organization

This thesis is organized as follows:

Chapter 2 discusses in a more detailed way the MST problem by making an overview and exposing some tracking methods, apart from this a real example will be introduced in order to illustrate the problem we are dealing with.

Chapter 3 will introduce some machine learning topics that we consider essential for the reader to know, ranging from feature selection techniques to prediction models definition as well as some information about how to validate those models.

In Chapter 4 the related work regarding the use of machine learning techniques over MST problem will be discussed in detail and some examples of the advances done so far will be given.

Chapter 5 will expose in detail the methodology we have designed in order to achieve our goal: specific problems and challenges that have arisen will be presented as well as how have they been addressed.

Chapter 6 will be centred in explaining how have we validated and tested our methodology: the different test and validation datasets will be introduced and their results will be discussed deeply.

Chapter 7 will be about conclusions and future work: in this chapter we will discuss what conclusions can be extracted from the whole process, analyzing whether our contribution has any relevance in MST problem or not. Apart from it the future

[1] Ichnaea was the tracing goddess, one of the female Titanes.

work for this thesis will also be introduced, mainly it has to do with a very promising new paradigm on MST problem.

Chapter 2

Microbial source tracking

In this chapter the concept of MST will be discussed in a more detailed way by making an overview as well as by exposing some tracking methods. Finally, a real example will be given in order to exemplify the problem we are dealing with.

2.1 Overview and relevance

By definition, MST is an emerging sub-discipline of Biology that allows practitioners to discriminate among the many possible sources of fecal pollution in environmental waters, for instance the water that a river contains. MST applications range from beach monitoring to total maximum daily load assessment of pollution sources, that is, whatever scenario in which public health is prone to be affected or that has potential capacity to improve environmental water quality and, therefore, increase the protection of public health.

Focusing on public health, citizens of developed and industrialized countries are generally protected by their own countries in the sense that the governments are responsible for the water quality that is used for several purposes such as drinking, personal hygiene, agriculture water, food production, and so on; this means that citizens of developed countries are protected against diseases caused by contaminated water. However, developing countries do not have, unfortunately, the necessary resources in order to maintain a minimum quality on their water resources with the aim of reducing microbial contaminants. Having this in mind, it is easy to realize that MST has a tremendous impact since the definition and implementation of microbial indicators to assess reductions in microbial pathogens of fecal origin has been proved to be an efficient way for the protection and improvement of water resources.

Pathogens from infected animals or humans can be introduced into water through feces or sewage and can cause serious human health risk, therefore, the identification of animal fecal sources is important to protect humans from zoonotic pathogens that can have their origin in birds, poultry, cattle and pigs. What is more, the capability to detect human-originated pollution is also extremely important since sewage from human origin is generally expected to have a higher risk to public health than animal origin.

2.2 MST history

The history of MST can be divided into several periods:

The initial one was around the 90's and basically was centred on the definition of new indicators (Brown 1993; Awad-El-Kariem et al. 1995; Hsu et al. 1995; Tartera et al. 1989; Bernhard and Field 2000; Nebra et al. 2003) and appropriate methods for fecal source discrimination (Hagedorn et al. 1999; Wiggins 1996; Parveen et al. 1997; Whitlock et al. 2002; Harwood et al. 2000; Manero et al. 2002; Wallis and Taylor 2003). The second period started in 2003 and it had to do with three large multilaboratory studies with the aim of comparing the different used methods for MST problem; apart from it the number of workshops, book chapters and articles dedicated to summarize the available information on the topic grew up a lot (Field et al. 2003; Harwood et al. 2003; Griffith et al. 2003; Myoda et al. 2003; Noble et al. 2003; Ritter et al. 2003; Blanch et al. 2004; Blanch et al. 2006). Moreover, a US federal guide that described the uses and limitations of MST methods was published (US Environmental Protection Agency 2005) as well as a book dedicated to MST as an emerging issue in food safety (Santo Domingo and Sadowsky 2007).

Over the last years, library-dependent tracking methods (the ones that require a large assembly of microorganisms from several host sources) have been replaced by library-independent tracking methods, that rely on detection of a particular host specific organism or gene. Next section will explained in a more detailed way both dependent and independent tracking methods.

To date, there has been no general consensus among researchers or any regulatory agency about what are the best indicators for determine the fecal origin in water. Many of the current studies are still focused only on the development of new MST indicators and the improvement of detection and quantification; however, another branch of research exists and it has to do with the diversity of MST approaches and new methodologies with different organisms.

2.3 Tracking methods

2.3.1 Library-dependent

A range of bacterial source tracking techniques is grouped under what is commonly called library-dependent methods (LDM). These methods require the construction of a library of known source profiles of fecal isolates from different animal sources that can be used to compare only with the same profiles of isolates from the environment to determine the origin of fecal contamination. This approach is based on the hypotheses that certain characteristics of fecal bacteria are associated with specific animals, that these characteristics in environmental strains are similar to those found in hosts groups and that the relative proportion of the identifying characteristic remains constant in the environment over time.

As the determination of fecal pollution source is based on the library, success of the methods depends on how well the library has been built paying special attention to distribution of fingerprint patterns amount the potential sources, representation of each one of the sources and the method of statistical analysis. One great advantage of LDM's is that the library can be adapted exclusively to a particular waterbody.

2.3.1.1 Representativeness and proportionality

In order to classify the source of environmental water isolates is extremely important that the library is large enough to contain a sufficiently diverse set of profiles that cover all the possible animal sources in an specific watershed.

Is has to be taken into account that for a six-class classification (six different pollution sources) there might be equals numbers of examples from each source; however, in order to distinguish human from non-human origin the distribution becomes biased and creates some challenges in the statistical analysis that would lead to use some particular statistical models instead of others.

2.3.1.2 Geographic stability

Geographic stability is another of the factors that must be taken into account when building and using a library. This is because some microbial behaviour patterns vary sufficiently enough to be not representative for other locations: for instance, a library developed in UK would not be representative at all for other countries like France, Sweden and Spain.

This suggests that libraries may need to be developed more locally for smaller geographic areas such as a US state or between Australia catchments within a 100-km radius; despite some accuracy can be lost the different libraries of each one of these areas can be merged with enough guarantees. However, for larger geographic areas (different states in the US, different countries in Europe) separate libraries need to be developed for each one of the regions.

2.3.1.3 Temporal stability

Temporal stability of a library refers to how stable a library is over a period of months or years, generally a library is developed and built to be used over a period of several months or years.

Not all the possible indicators behave in the same way as time passes, for instance: *E. Coli* indicator organism for source tracking has raised some concerns in the sense that there is little stability in the clonal composition of populations in individual hosts, host populations, and locations over periods as short as some weeks; on the other hand, libraries of *enterococci* may be more stable since those that are composed of antibiotic resistant profiles have been reported to be as stable as 1, 3 and 5 years.

It has been suggested that both temporal and geographic stability can be related with the discriminatory power of the prediction methods used, thus, those factors must be taken into account when developing a library since the degree of success of a methodology will be seriously affected by them.

2.3.1.4 Summary

Library-dependent methods have a set of advantages particularly in studies where the relationship to fecal indicator bacteria and categorization of a number of sources is needed for the development of strategies that lead to reduce the impacts of contamination. However, LDMs can be costly and time consuming because of the library

building process they involve, that usually includes isolating and culturing isolates from water sampling.

Concluding, the construction of a representative library requires taking into account the different types and diversity of sources in the watershed and the careful selection of an appropriate method as well as testing the library performance.

2.3.2 Library-independent

Routine detection of fecal pollution is still based on fecal indicator bacteria (FIB) including *E. coli* and *intestinal enterococci*, water-quality testing based on the application of standard FIB has contributed to a fundamental improvement in water safety management during the last century. There are several reasons to use bacterial cells when detecting the origin of fecal pollution:

- bacteria are highly abundant in fecal materia.
- sampling them is relatively easy.
- detection methods are well established.
- there is no particular risk for the laboratory personal during analysis.

E. coli and *intestinal enterococci* are considered as indicators of fecal pollution origin since they occur in human an animal fecal pollution sources. Methodologies based on these indicators require library-dependent MST methods (see last section), on the other hand, bacterial targets for library-independent MST methods have to be source-specific.

2.3.2.1 Basic requirements

There are some basic requirements that the ideal MST methods should meet:

- Source-specificity criterion: bacterial MST targets should only be present in the fecal material of the considered source group, this means the target should be absent in the fecal material of all other source groups.
- Source-sensitivity criterion: MST targets should be present in comparable numbers in the feces of all subgroups of targeted sources.
- Source-group comparability criterion: knowledge on quantitative target occurrence in each of the source groups is required if the significance of fecal pollution among source groups is going to be compared; what is more, information on the environmental persistence and proliferation of the target is also required due to the fact that some targets may remain undetectable under certain circumstances while others tend to persist for a prolonged periods of time.

In this whole section we have been using two terms that maybe need to be clarified: a fecal *source group* might be for instance a specific animal species, a larger group of species sharing specific traits or a even broader group, like mammals; in this sense is essential to clearly define the boundaries of a targeted source group. On the other

hand, the *target* of a library-independent MST method can be either source-specific bacterial gene, a host-associated bacterial population or a bacterial community; most of the current MST method belong to the host-associated populations group.

2.4 Modelling concentration level and environmental persistence

This section will introduce the concepts of *concentration level* and *time persistence* when measuring a set of indicators from a waterbody. We will also explain how these two processes can be modelled numerically.

2.4.1 Concentration level

Concentration level is the relationship among the amount of particles of an indicator and the volume of water in which it has been measured. For example: let m be the measurement of some indicator within a volume v; if v is doubled and then a measurement is done it will be $\frac{m}{2}$ instead of m.

The concept of *concentration level* leads to the concept of *dilution*, lowering the first one implies increasing the second one. On last example we were doubling the *dilution factor* by halving the *concentration level*.

Dilution is very easy to manage numerically: let m be the numerical measurement of an indicator, if we want to dilute m by a factor of d the result will be simply $\frac{m}{d}$.

2.4.2 Time persistence

Modelling how a particular indicator persists in time is not as easy as modelling how it is diluted, mainly due to two reasons:

- Almost all the indicators persist in time in a different way.
- Given one particular indicator, it will persist in time in a different way depending of the season of the year.

This means that the only way of knowing how an indicator persist in time is by having several measurements on it along the time for several seasons of the year. Once these measurements are provided, the *time persistence* of an indicator can be modelled, for instance, by calculating a linear regression on them.

Figure 2.1 shows several measurements on FC (enumeration of fecal coliforms) indicator along the time as well as a linear regression performed on them. X-axis represents time (in hours) while Y-axis is the log of FC.

The actual simple linear regression could be used to model the way the indicator persists in time for a particular season, as follows:

$$\tilde{v} = v + at$$

where:

- $\tilde{v}$ is the new *aged* value.

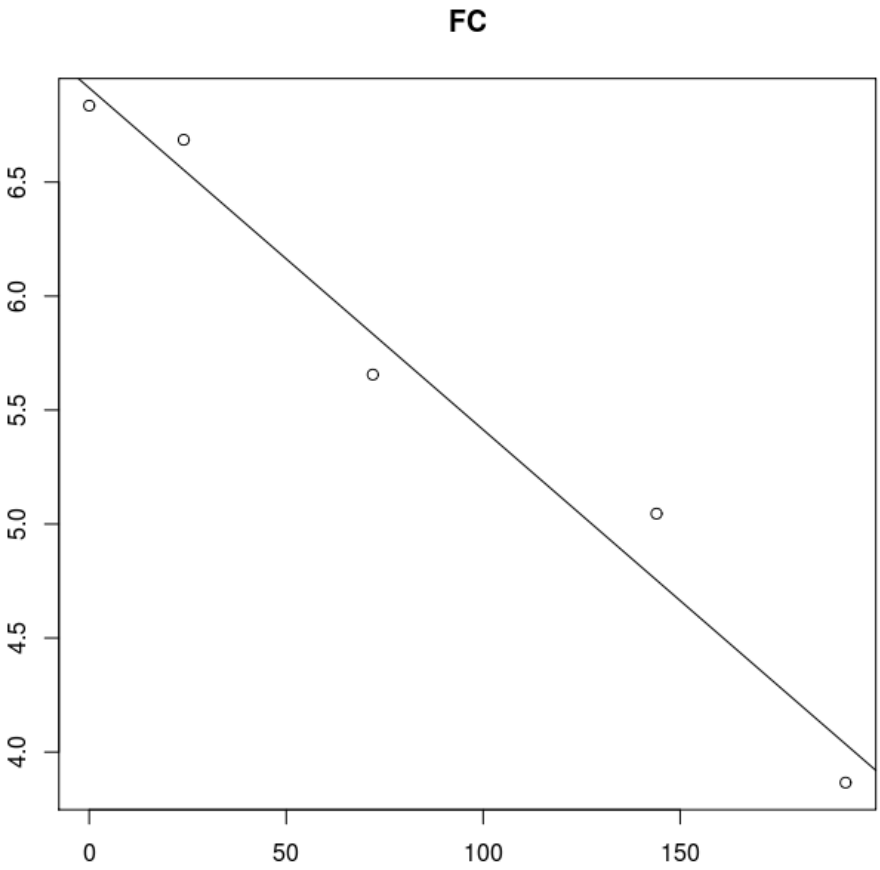

Figure 2.1: Several measurements on FC indicator along the time. A linear regression on them is shown.

- v is the *point of source* measured value.
- t is the elapsed time.
- a is the slope of the linear regression over the indicator time measurements.

We have selected linear regression because it models very good the log of the indicators; this is actually the way biology practitioners do it [4].

2.4.3 Detection threshold

Each one of the indicators within an MST study has a *detection threshold* below which it remains undetectable.

As long as a indicator is *diluted* and/or *aged* its value is more and more diminished. There is a point in which the indicator arrives to what is called *detection threshold*. At that point the indicator can be no more *diluted* nor *aged*.

It is because of the *detection threshold* that *diluting* and *ageing* a sample makes harder the problem of separating *human* from *non-human* origin.

2.5 A real example

In this section an example with real data will be shown by plotting a population of 103 samples using two indicators, more precisely, one indicator and the ratio formed by this indicator and the other one. Each sample contains more indicators but just two of them will be shown for visualization simplicity. The two indicators that will be used are:

- Enumeration of *somatic coliphages*, abbreviated as SOMCPH.
- Enumeration of *B. fragilis* bacteriophages using the new host strain *B. thetaiotaomicron*, abbreviated as GA17.

As has been said before, one of the indicators used on the plot will be SOMCPH itself while the second one will be ratio formed by SOMCPH and GA17, obtained by the simple division

$$SOMCPH/GA17$$

In these 103 samples the aim is just distinguishing among human and non-human samples, thus the class of each one of the samples will be represented by a different symbol.

2.5.1 Heavily polluted water

Figure 2.2 shows 103 samples that have been taken from a heavily polluted waterbody, this means that this sample is not affected at all by different *concentration levels* or *environmental persistence*.

As can be seen on figure 2.2 the samples belonging to each one of the classes (human and non-human) are completely separable: this means that a hypothetical prediction model built over these data could be able to separate both classes by establishing the corresponding relationship among both indicators.

Imagine a line that separates both groups perfectly: this simple line would be a prediction model which is actually able to separate both classes. There are infinite ways of separating the sources on figure 2.2 and some of them are definitely better than other ones. According to statistical learning theory, the separation which leaves the maximum margin among the classes is considered the best one; in other words, the one that generalizes better [11].

These topics are out of the scope of this chapter and will be treated on the next one, in which several kind of models and their advantages and drawbacks will be discussed deeply.

2.5.2 Different concentration level and time persistence

We would like to conclude this chapter by showing the effects of lower *concentration level* (sample is said to be *diluted*) and higher *time persistence* (sample is said to be *aged*); in other words, when the sample is not taken at heavily polluted (*point of source*) waterbody.

Figure 2.3 shows the same 103 samples as figure 2.2 but with quite lower *concentration level*, several observations can be done over this plot:

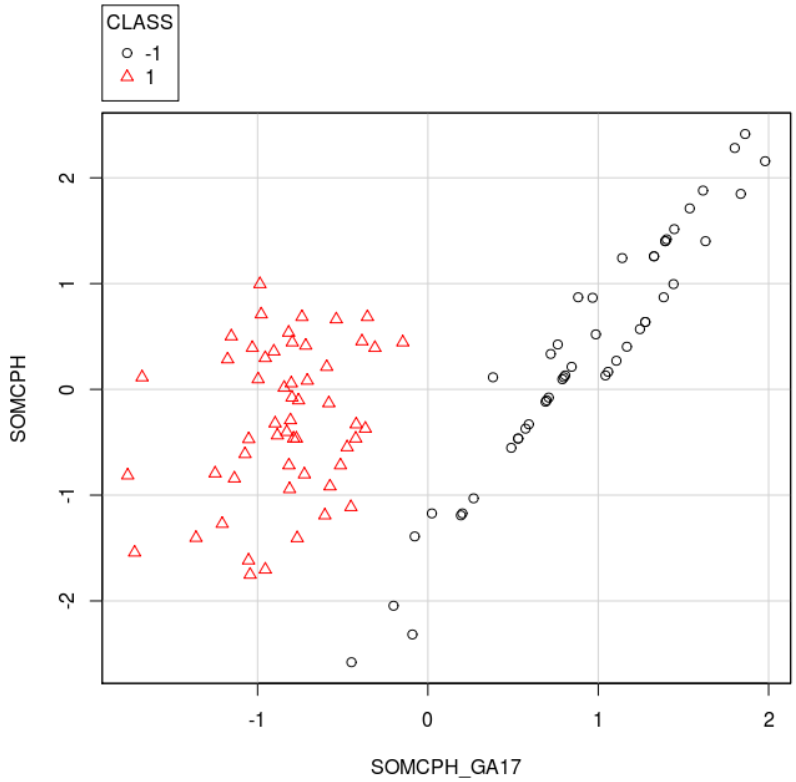

Figure 2.2: Distribution of samples according to indicators SOMCPH and SOMCPH/GA17 on heavily polluted water.

- Human and non-human sources are still separable but with a smaller margin than in figure 2.2.
- Having lower *concentration level* makes detecting fecal origin a harder task.
- A prediction model built over this data is not as reliable as before. Since the margin for separate both classes is smaller, the ability to *generalise* data is diminished.

Figure 2.4 shows the same 103 samples as figure 2.3 but with higher *time persistence* (besides the lower *concentration level*), several observations can be done over this plot:

- Human and non-human sources now are not separable since both classes morphology have become overlapped.
- Thus, the problem of separating both classes has become even harder.
- Having higher *time persistence* makes detecting fecal origin an even harder task.
- A prediction model built over this data is not reliable at all.

We have two indicators (SOMCPH and SOMCPH/GA17) that are excellent discriminators when they are taken at *point of source*.

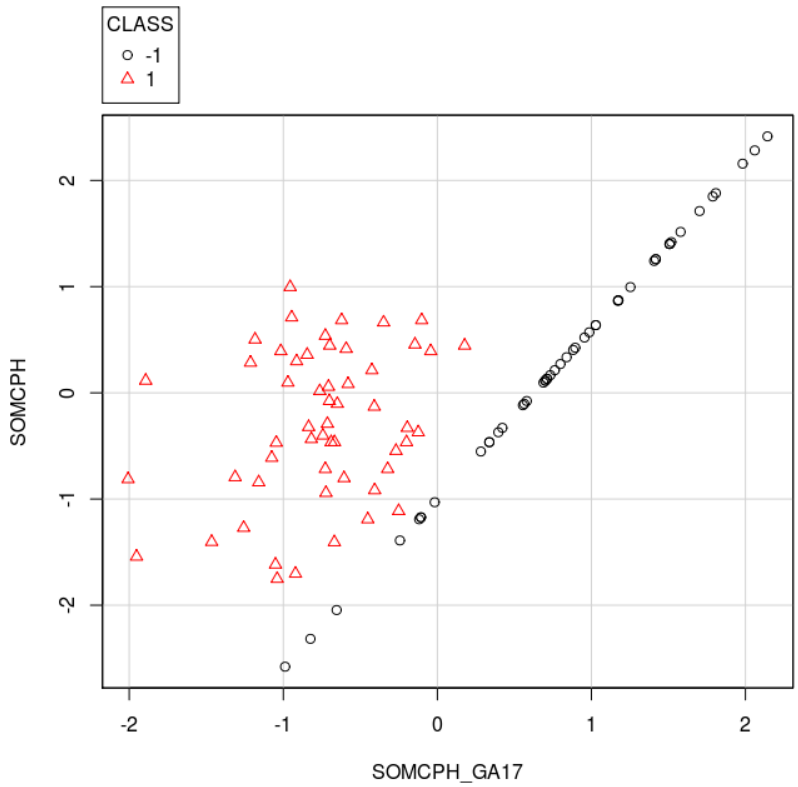

Figure 2.3: Distribution of samples according to indicators SOMCPH and SOMCPH/GA17 showing low concentration level.

However, both indicators start to lose their effectivity under a low *concentration level* situation. The effectivity is completely lost when the *time persistence* is increased.

The bottom line of it is that, despite some indicators are excellent when measured at the *point of source*, this does not guarantee that they are going to be also good at low *concentration levels* or high *time persistence*.

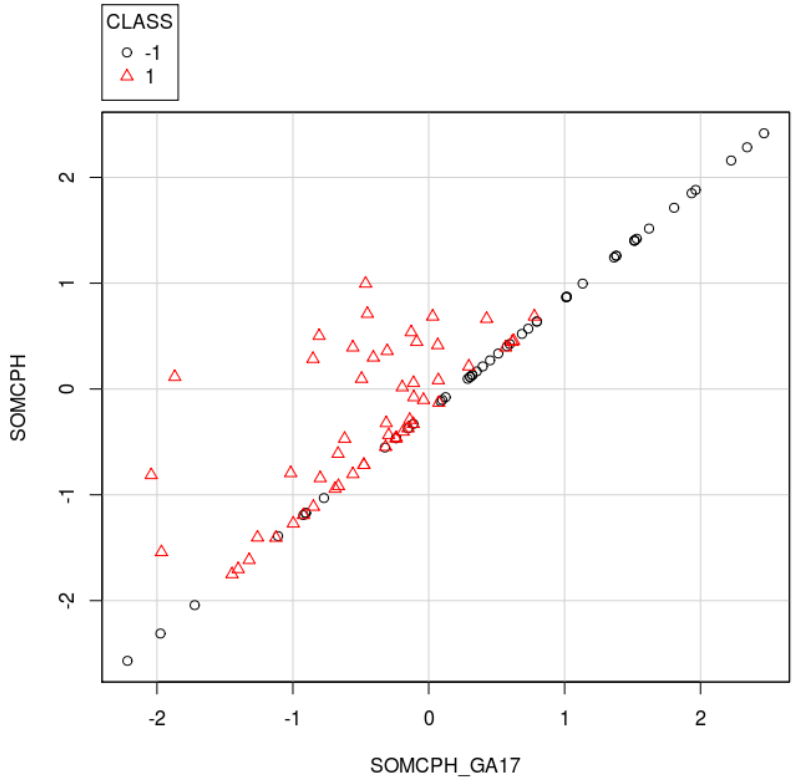

Figure 2.4: Distribution of samples according to indicators SOMCPH and SOMCPH/GA17 showing low concentration level and high time persistence.

Chapter 3

Learning methods and techniques

This chapter will cover several topics within machine learning that has been used in this research, from its very basic definition and overview to some of its techniques such as *feature selection* and *validation protocol* as well as some prediction methods (models) that are also used in our methodology, particularly:

- Discriminant analysis.
- Support vector machines.
- K-nearest neighbors algorithm.
- Logistic regression.

3.1 Overview

Machine learning (ML) is nowadays one of the most important branches of artificial intelligence, its main aim is to develop methods an algorithms that allow computers to learn or to extract knowledge or behaviour patterns based on empirical data, which can be obtained from several sources such as databases, sensors, surveys, etc.

An learning algorithm or model is usually fed with several (from tens to thousands) examples (data) and it is able to extract or infer the characteristics of interest of this data. The challenge comes up due to several facts:

- There is no knowledge of the underlying probability distribution that has generated the data.
- The given examples are only a sample of the whole population.
- The model or algorithm should be able to generalize just from the given examples in order to be useful and reliable when new unseen examples are coming.

By generalizing we mean the ability of performing in a correct way when analyzing new examples that have not been seen during the training phase. This means not to focus just on the given examples and memorizing them but being able to learn or extract something more general.

Within ML there two big groups of algorithms: those that perform *supervised learning* and, on the other hand, those that perform *unsupervised learning*. In *supervised learning* each one of the given training examples are labeled with their desired output, learning is called supervised because the model or learner will know during the training phase the desired output for the current example; on the other hand, in *unsupervised learning* the examples are not labelled at all and usually the main algorithm task is finding groups or clusters formed by subsets of the examples that have similar characteristics. Apart from them also exists the so called *reinforcement learning*, which has to do with agents that make some decisions in a particular environment in order to maximize some kind of reward, whatever this reward is; more details over this last kind of learning will be given in next sections.

3.2 The learning problem

3.2.1 General formulation

Learning is the process in which the unknown dependency between an input and an output is estimated using a limited sample of observations. Learning scenario (see figure 3.1) involves three components:

- *Generator of samples*: produces independent random vectors $\mathbf{x} \in \Re^d$ from a fixed and unknown probability density $p(\mathbf{x})$.
- *System*: produces an output value y given an input vector $\mathbf{x}$ according to a fixed and unknown conditional probability density $p(y|\mathbf{x})$. This formulation includes both *deterministic* system, where $y = t(\mathbf{x})$, and the *stochastic* one, where $y = t(\mathbf{x}) + \xi$ being ξ random noise with zero mean.
- *Learning machine*: implements a set of functions $\hat{y} = f(\mathbf{x}, \omega)$, $\omega \in \Omega$ where Ω is the set of abstract parameters that characterize the set of functions.

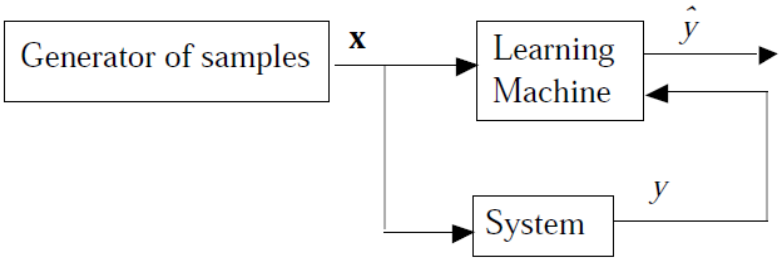

Figure 3.1: General learning scenario (extracted from [7])

The problem that the *Learning Machine* has to address is selecting a function (from the set of functions it supports) that best approximates the *System*'s response just by observing a finite number (n) of independent and identically distributed examples produced by the *Generator* and the *System* according to the unknown joint probability density function

$$p(\mathbf{x}, y) = p(\mathbf{x})p(y|\mathbf{x}).$$

The quality of the selected function from the *Learning Machine* is measured by the loss (or discrepancy) function $L(y, f(\mathbf{x}, \omega))$ between the output produced by the *Learning Machine* and the *System* for a given input $\mathbf{x}$, this loss function takes on non-negative values so that large positive values correspond to poor approximation. Loss function expected value is called the *risk functional* and is defined as:

$$R(\omega) = \int L(y, f(\mathbf{x}, \omega))p(\mathbf{x}, y)d\mathbf{x}dy.$$

Having that, learning can be defined as the process of estimating the function $f(\mathbf{x}, \omega_0)$ that minimizes $R(\omega_0)$ using only a finite sample of examples given that $p(\mathbf{x}, y)$ is completely unknown. Since we just have a finite set of examples, which is a sample of the whole population, we cannot expect to find the optimal solution $f(\mathbf{x}, \omega_0)$ but just an approximation of it, so we denote $f(\mathbf{x}, \omega^*)$ as the estimation of the optimal solution obtained with a finite set of examples using some learning method.

Application over a finite set of examples

Let $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be a set of N examples being $x_i \in \Re^m$ the input of the i-th example, whose desired output is $y_i \in \Re^p$.

The learner objective is to find a function $f : \Re^m \times \Omega \rightarrow \Re^p$ with the aim of modelling the relationship between the input and the output spaces. In order to quantify in some way how good does f models the relationship between both spaces a loss function $L : \Re^p \times \Re^p \rightarrow \Re^+ \cup \{0\}$ can be defined as $L(y_i, f(x_i, \omega^*))$.

Having defined the loss function the *empirical error* of the N samples can be represented as:

$$R(\omega^*) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i, \omega^*)).$$

The regularization approach

Once $R(\omega^*)$ function is defined the main strategy of any kind of model would be to minimize this function to make function f as close as possible to the real (and completely unknown) function that establishes the relationship between the inputs and the outputs of the given example.

However, it is definitely not a good strategy to minimize $R(\omega^*)$ towards zero (in fact, some models like *neural networks* are potentially able to do it) because this means that the model is not *generalizing* at all the given input data nor extracting any knowledge, instead of it what the model is doing is just *memorizing* the data just as if it was a look-up table.

There are several strategies to avoid this behaviour and allowing the *learning machine* to *generalize* instead of *memorizing*: one one this strategies is the so called *regularization* which consists in adding to the function to be minimized a weighted term (called *penalty term*) that measures the complexity of the *learning machine*.

In order to achieve this a new penalized risk functional will be defined as:

$$R_{pen}(\omega^*) = R(\omega^*) + \lambda\phi[f(\mathbf{x}, \omega^*)]$$

where:

- $\lambda \in \Re^+ \cup \{0\}$ is the regularization factor, the higher it is the more the *learning machine* complexity will be penalized.
- $\phi[f(\mathbf{x}, \omega^*)]$ measures the complexity of the *learning machine* $f(\mathbf{x}, \omega^*)$.

Under this approach we are expected to find the optimal model estimate (optimizing λ by the use of resampling methods) as a result of a tradeoff between fitting the data and a prior knowledge about the complexity of the *learning machine*.

3.2.2 Supervised learning

In *supervised learning* each of the given training examples is labeled with its desired output, learning is called supervised because the model or learning machine will know during the training phase the desired output for the current example.

The learning machine is intended to analyze data and to infer a function, that will be called a *classifier* if the desired output is a discrete set of possible values or a *regression function* if the output is continuous.

Supervised learning is the only kind of learning that has been used on our methodology since all the examples of our training data are always labelled with the desired output, what is more, our problem is classification problem with classes, hence we can consider the *learning machines* our methodology is going to build as *classifiers*.

Whether if we are in a context of a continuous desired output or, on the other hand, we are dealing with a discrete set of desired output we will need different loss functions depending on the case, let us define next what would be the discrete loss functions in each one of the two cases that have just been mentioned:

Loss function for classification

Loss function for classification problems just takes into account if the output given by the *learning machine* is exactly the same as the desired output. In that case, the returned value of the loss function will be 0.

However, if the output given by the *learning machine* is different from the desired output the error will be mapped by the loss function by returning a 1 (or any positive number).

$$L(y_i, f(x_i, \omega^*)) = \begin{cases} 0 & \text{if } y_i = f(x_i, \omega^*) \\ 1 & \text{if } y_i \neq f(x_i, \omega^*) \end{cases}$$

Loss function for regression

Regarding with the loss function for regression problems it must be pointed out that there are many kinds of error functions for regression problems. At this point we are going to introduce the squared error function since is a well-suited and widely used error function for regression.

However, squared error function suffers when dealing with outliers (isolated data examples that are far from the desired output value) in the sense that they are heavily punished by the squaring of the error.

$$L(y_i, f(x_i, \omega^*)) = (y_i - f(x_i, \omega^*))^2$$

3.2.3 Unsupervised learning

Unsupervised learning involves learning patterns in the input when no specific output values are supplied, therefore the actual problem is discerning multiple categories in a set of examples. The problem is considered *unsupervised* since the category labels of each one of the examples are not given.

As an example, one of the main approaches in *unsupervised learning* is the cluster analysis: which is the task of assigning a set of examples into one or more different groups according to a notion of similarity. There are several algorithms in order to perform a cluster analysis into a set on unlabelled examples being one of the most important the *k-means* algorithm: in which a set of n examples is clustered into k clusters (or groups) being k a parameter of the algorithm.

3.2.4 Reinforcement learning

Reinforcement learning is an area of machine learning in which a *learning machine* makes decisions in a given environment based on a kind of feedback called *reward* or *reinforcement*. Active *reinforcement learning* model includes:

- A set of states S that describe the environment.
- A set of actions A.
- Rules that define the transitions among the states.
- A reinforcement function that tells the agent the results of an action performed on the environment.

The goal of a *reinforcement learning machine* is to collect as much reward as possible, in this sense the agent chooses the actions according with the history of collected rewards up to that moment. It must be pointed out that in the first stages of the learning process the agent can even randomize the action selection.

3.3 Feature selection

3.3.1 Overview

Feature selection (FS) is the process of selecting a reduced set of attributes from all the attributes that belong to the data attribute set. There are several reasons for approaching in FS and not to use directly all the attributes that we are given:

- Allows to reduce the overall size of the data by dropping irrelevant and redundant variables.
- Good FS can improve the quality of the *learning machine* since it can focus only on relevant attributes.

- Good FS can also express the resulting model as a function of few variables, making it much more understandable.

- Dimensionality reduction to 2 or 3 attributes may be required so that data can be graphically represented.

At first sight, it seems better to have a lot of attributes than having just a few of them because we will have more information about the problem and thus we will be able to build a better *learning machine*; however, what will happen is that many learning methods will get lost within this high dimensionality space (specially if there are relevant and redundant attributes or some of them contain wrong values), the consequence of this will be that the *learning machine* is going to be prone to fit the particularities of data and will not generalize it at all.

Defining FS more formally, given an attribute set $X_n = \{x_1, \ldots, x_n\}$ the goal is finding a subset $X_m = \{x_1, \ldots, x_m\}$ being $m < n$, $X_m \subset X_n$, that maximizes an objective function $J(X_m)$ so that:

$$X_m = \{x_1, x_2, \ldots, x_m\} = \arg \max_{Y \subseteq X_m} J(Y)$$

Being $J(Y) : Y = \{y_1, \ldots, y_p\}$ a measure of the discrimination power of the attributes belonging to Y regarding with the response variable.

At a very first stage of a FS process some (usually no more than a few of them) can be dropped easily if, for instance:

- An attribute is constant (has the same value) for all the examples of the dataset.

- We are provided with external knowledge (usually an expert in the field the dataset is about) so that some attributes are redundant or irrelevant.

Once these attributes have been dropped, if we want to keep on reducing dimensionality we are required to use more sophisticated techniques like the ones that will be described next.

3.3.2 Wrapper approach vs. filter approach

Wrapper approach

In the *wrapper approach* to FS, the FS algorithm uses a search strategy within the space of possible feature subsets and evaluates each one of the subsets against a model m.

Therefore, in the *wrapper approach*, the objective function $J(Y)$ could be the error estimation, possibly using some kind of resampling method like cross-validation (see section 3.4.3), of a k-NN algorithm (see section 3.5.3) using attributes set Y.

The key thing within the *wrapper approach* is that the model m used to evaluate *objective function* J is the same inducer as the one used to build the final predictive model. According to the prior example in this case the model should be the k-NN algorithm (see section 3.5.3).

The main aim of the *wrapper approach* is evaluating the highest possible number of attributes subsets with the *objective function* and then establishing an attributes

ranking according to the value returned by this function. However, the crucial point in this approach is deciding which ones of attributes subsets do we evaluate because in the large majority of problems is completely unfeasible to evaluate all the possible combinations of attributes. Let us mention two possibilities:

- Depending on the total number of attributes it would be feasible to evaluate all the attributes combinations up to a limited attributes subset size, say 4.
- Using some heuristic or guided method in order to evaluate only the attributes subsets that potentially have good discrimination power.

The main drawback of the *wrapper approach* is that it is prone to overfit the model and that is computationally more expensive that the *filter approach*.

Our methodology uses a *wrapper approach* in order to perform feature selection, that will be explained in a detailed way later in section 5.

Filter approach

Besides the wrapper approach another feature selection technique exists and is called the *filter approach*.

The main difference between the *filter approach* and the *wrapper approach* is that while the last one evaluates the performance of a set against an specific model in the *filter approach* a simple filter is evaluated, this means that irrelevant attributes are filtered before engaging the learning process.

The large majority of filters are based on statistical measurements that are calculated from the data itself. Some *filter approach* techniques include the use of:

- *correlational analysis*: detecting relationships between attributes and check which ones of them seem to be more independent.
- *correspondence analysis*: a chi-squared test is performed in order to detect if the values of a variable are independent from other variables values.

3.3.3 A feature selection algorithm example

A very popular and widely used feature selection algorithm is the so-called *Sequential Forward Selection* (SFS), let us introduce the algorithm in order to exemplify how does a wrapper-approached algorithm works:

As can be seen on algorithm 1 it performs a kind of guided search throw the whole space of solutions, which is formed by all the existing sets $Y_k = \{y_1, \ldots, y_k\} \subset X = \{x_1, \ldots, x_n\}$ being $k < n$.

Performing the search on the whole solution space would be completely unfeasible for large values of n. For this reason the search is guided in the sense that at every step k the best attribute, according to the value returned by the objective or evaluation function J, from the ones that have not been added yet is added to the Y_{k-1} attribute set.

Algorithm 1 Sequential Forward Selection

Input: $X \leftarrow \{x_1, \ldots, x_n\}$
  $Y_0 \leftarrow \emptyset$
  $k \leftarrow 0$
  while $X \neq \emptyset$ **do**
    $y \leftarrow \arg\max_{x \in X} J(Y_k \cup \{x\})$
    $k \leftarrow k + 1$
    $Y_k \leftarrow Y_{k-1} \cup \{y\}$
    $X \leftarrow X \setminus \{y\}$
  end while
  return $\arg\max_{Y \in \{Y_0, \ldots, Y_k\}} J(Y)$

3.4 Validation protocol

Within this section several ways of assessing learning system's performance so that they can be compared will be introduced and their advantages and drawbacks will also be exposed.

In any learning process there is a point in which several models have to be compared so that the best (or the best ones) model/s can be chosen, in this sense there is a need of having a way of assessing (in the most possible reliable way) the quality of model or how good or bad it performs.

The main challenge comes from the fact that any learning process has a set of finite examples of data and both the training of the model and the assessment of its quality has to be done using just this data. Next subsections will expose several methods of splitting the available data in order to obtain good models whose quality is assessed in a reliable way.

All the methods below split the entire dataset D into three subsets:

- Training set (TR): examples belonging to TR set will be used to train the system.
- Validation set (VA): examples belonging to VA set will be used to optimize model's parameters by calculating the *validation error*, this is the measure that will be used to compare models
- Test set (TE): once the best model is selected using the *validation error* the TE set is used to assess how good (or bad) is the best model that have been selected by calculating the *test error*. Examples belonging to TE set must not be used during the training and validation processes.

Thus, before applying any of the methods below a TE set must be separated from the entire data in order to use it towards the final step of the whole learning process for assessing how good is the best final model that we have selected. For clarifying purposes let us define the set $D_{TV} = TR \cup VA$ being $D_{TV} \subset D = TR \cup VA \cup TE$.

3.4.1 Hold-out

Hod-out is by far the simplest of all the validation methods: it simply consists in splitting D_{TV} into two separate sets TR and VA being usually $|TR| >> |VA|$; then

the *learning machine* is trained using the TR set and its *validation error* is assessed using the VA set.

This technique is suitable and should only be used if the entire dataset D is large enough so that TR and VA are representative for the variance of the data. Otherwise, if D set is not large enough the *learning machine* will have a reduced amount of data for both training and validation and it will incur into large variance: this means that the *learning machine* may not be bad for the current data but if the data changes its results will vary a lot since it has not been able to generalize the data in a proper way.

3.4.2 Random sub-sampling

One way to manage the great amount of variance of the hold-out approach when the entire dataset is not large enough is by using the repeated hold-out approach, that consist in:

- Apply the hold-out method n times to produces different splits of the D_{TV} set.
- In each iteration, TR and VA sets should be selected randomly.
- Final *validation error* is the mean of the n calculated validation errors.

This method improves the simple hold-out presented in last section; however, there is plenty of room for improvement since overlapping among the sets (both TR and VA) of different iterations can occur. This would not be a problem in large datasets but in small ones it is.

3.4.3 K-fold cross-validation

One more step in order to solve the problems of the prior presented method lead us to introduce the *k-fold cross-validation* approach, whose main aim is avoiding the overlapping among sets and that works in this way:

- D_{TV} is splitted into k disjoint subsets (folds) $\{D_{TV1}, \ldots, D_{TVk}\}$ of approximately equal size.
- At the i-th iteration D_{TVi} is used as the validation set while $\bigcup_{j \neq i} D_{TVj} : j = 1..k$ is the training set.
- Final *validation error* is the mean of the k calculated validation errors.

This approach is specially suited when the data set is not very large so that an accurate *learning machine* performance estimation is obtained. Typical choices for k value are 10 or 5 but some considerations regarding with the value of k must be taken into account:

- The larger the k value is, the larger will be the TR set and the smaller will be the VA set, this means a less biased performance estimation but with a higher variance, and vice-versa.
- *K-fold CV* means that k learning machines will be built; hence, the higher k is, the more computing time is required.

3.4.4 Leave-one-out cross-validation

Leave-one-out cross-validation (LOOCV) is a special kind of *cross-validation* approach in which $k = |D_{TV}|$. This way of choosing the folds implies that there is just one way of splitting D_{TV} if LOOCV is used.

LOOCV evaluation method provides an unbiased error measure but with a high variance and it is specially suited for small or very small datasets.

3.5 Prediction models

3.5.1 Stabilised linear discriminant analysis

Linear discriminant analysis (LDA) is a method used in statistics and machine learning. Its objective is finding a linear combination of the variables which characterizes or separates two or more classes. The resulting combination is called *linear classifiers*.

Let us assume we have a sample of observations $D_i = \{x_1^i, \ldots, x_n^i\},\ x \in \Re^p,\ i \in \{1, \ldots, C\}$ from class ω_i.

Discriminant analysis is based on the normal distribution:

$$p(x|\omega_i) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\}$$

Classification is achieved by assigning a pattern to a class for which posterior probability, $p(\omega_i|x)$ is greatest, or equivalently $\log(p(\omega_i|x))$.

Using Bayes' rule we have:

$$\log(p(\omega_i|x)) = \log(p(x|\omega_i)) + \log(p(\omega_i)) - \log(p(x)) =$$

$$-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - -\frac{1}{2}\log(|\Sigma_i^{-1}|) - \frac{p}{2}\log(2\pi) + \log(p(\omega_i)) - \log(p(x))$$

Since $p(x)$ is independent of class, the discriminant rule is: assign x to ω_i if $\forall j \neq i$, $g_i > g_j$ where:

$$g_i(x) = \log(p(\omega_i)) - \frac{1}{2}\log(|\Sigma_i^{-1}|) - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)$$

Since values μ_i and Σ_i are unknown they have to be estimated using sample D_i as follows:

$$\hat{\mu}_i = \frac{1}{n}\sum_{j=1}^{n} x_j^i$$

$$\hat{\Sigma_i} = \frac{1}{n}\sum_{j=1}^{n}(x_j^i - \hat{\mu}_i)(x_j^i - \hat{\mu}_i)^T$$

Problems can arise in the Gaussian classifier if any of the matrices $\hat{\Sigma_i}$ is singular. An alternative to solve that is to assume that the class covariance matrices $\Sigma_1, \ldots, \Sigma_C$ are all the same. In this case the discriminant functions becomes linear and de discrminant function is:

$$g_i(x) = \log(p(\omega_i)) - \frac{1}{2}\mu_i^T S_W^{-1}\mu_i + x^T S_W^{-1}\mu_i$$

where

$$S_W = \sum_{i=1}^{C} \frac{n_i}{n} \hat{\Sigma}_i$$

According to [17], Stabilised Linear Discriminant Analysis (SLDA), implements the LDA for q-dimensional linear scores of the original p predictors derived from the PC_q rule by Laeuter et al. (1998). Based on the product sum matrix

$$W = (X - \bar{X})^T (X - \bar{X})$$

the eigenvalue problem $WD = diag(W)DL$ is solved. The first q columns D_q of D are used as a weight matrix for the original p predictors: XD_q. More details on SLDA can be found at [17].

3.5.2 Support vector machines

General formulation (separable data)

The support vector machine (SVM) is learning procedure base on statistical learning theory [15].

Intuitively, given a set of examples belonging to two classes, the SVM tries to find the optimal hyperplane that separates them: this is the hyperplane that maximizes the distance from it to the nearest point of each class (this distance is called *margin*).

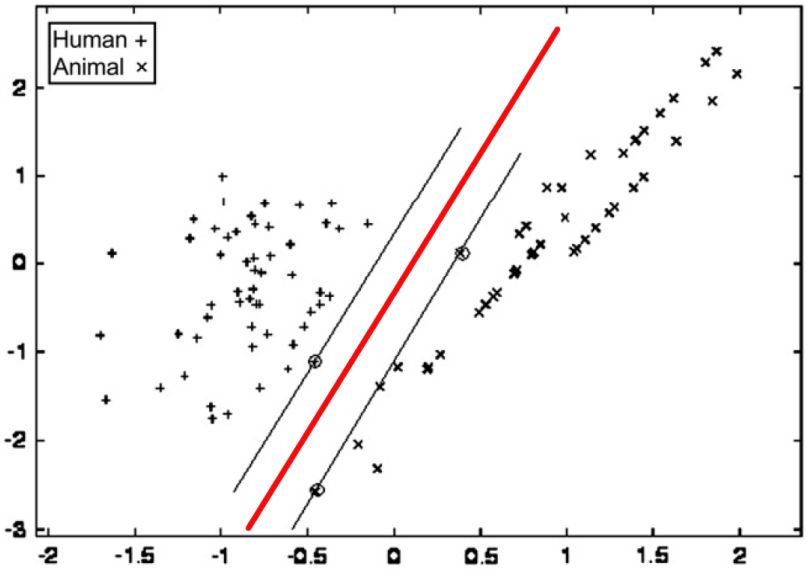

Figure 3.2: Support vector machine optimal hyperplane.

Figure 3.2 shows the optimal hyperplane in red colour (solid line in B/W paper) that separates both classes. The nearest examples of each class to the optimal hyperplane are indicated by a circle (they are called *support vectors*). This hyperplane is expected to generalize better by reducing the risk of misclassifying unseen examples.

Let $X = \{(x_1, y_1), \ldots, (x_n, y_n)\} : x \in \Re^d,\ y \in \{+1, -1\}$ be n **separable** samples used as training data.

Let $D(x) = w^T x + b$ be the hyperplane decision function that separates both classes.

Let $\triangle$ be the **minimal** distance from the separating hyperplane to its closest data point. Then, a separating hyperplane with margin $2\triangle$ holds that:

$$w^T x_i + b \geq \triangle \text{ if } y_i = +1,\ i = \{1, \ldots, n\}$$

$$w^T x_i + b \leq \triangle \text{ if } y_i = -1,\ i = \{1, \ldots, n\}$$

or more compacted:

$$y_i[w^T x_i + b] \geq \triangle,\ i = \{1, \ldots, n\},\ y \in \{+1, -1\}$$

For a given training dataset, all possible $\triangle$-separating hyperplane can be represented by the last expression.

However, a $\triangle$-separating hyperplane is called *optimal* if the margin is the maximum size allowed by the data. For a margin $2\triangle$, all training samples are at least $\triangle$ away from the decision boundary, so they hold that:

$$\frac{y_i[w^T x_i + b]}{||w||} \geq \triangle,\ i = \{1, \ldots, n\},\ y \in \{+1, -1\}$$

Therefore, the optimal separating hyperplane can be found by minimizing

$$\frac{1}{2}||w||^2$$

subject to

$$y_i[w^T x_i + b] \geq 1,\ i = \{1, \ldots, n\},\ y \in \{+1, -1\}$$

This can be solved by *quadratic programming* techniques that, in this case, imply:

- Quadratic function subject to linear constraints.
- Unique (or a set of equilavent) solution.

Extension to non-separable data

When the training data samples are not separable a set of slack variables to allow the possibility of examples misclassification, therefore, the optimal separating (allowing misclassifications) hyperplane can be found by minimizing

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i$$

subject to

$$y_i[w^T x_i + b] \geq 1 - \xi_i,\ \xi_i > 0,\ i = \{1, \ldots, n\},\ y \in \{+1, -1\}$$

being:

- $C > 0$ the *regularization* parameter: larger value assigns a higher penalty to misclassification errors.
- ξ_i slack variables, misclassified examples have $\xi_i > 1$.

Non-linear SVM

This section will discuss how the above method can be generalized to the chase when the decision function is not a linear function of the data [12].

This generalization can be done by mapping the data into a higher dimensional space (called *feature space*) $\mathcal{H}$ as follows:

$$\Phi : \Re^d \rightarrow \mathcal{H}$$

SVM training algorithm involves using the dot product of data $x_i \cdot x_j$ within the original space; therefore, in the *feature space* the training will depend on the data through dot products in $\mathcal{H}$ in the form $\Phi(x_i) \cdot \Phi(x_j)$.

Having that, if a *kernel function* K in the form $K = \Phi(x_i) \cdot \Phi(x_j)$ eventually existed, we would only need to use K in the training algorithm, and would never need to explicitly even know what Φ is, for instance:

$$K(x_i, x_j) = e^{\frac{-||x_i - x_j||^2}{2\sigma^2}}$$

In this particular example, H is infinite dimensional, so it would not be very easy to work with Φ explicitly. However, if one replaces $x_i \cdot x_j$ by $K(x_i, x_j)$ everywhere in the training algorithm, it will produce an SVM which lives in an infinite dimensional space, and furthermore do so in roughly the same amount of time it would take to train on the un-mapped data.

The trick is not to compute Φ explicitely but computing *kernel function* K instead of it.

Some examples of *kernel functions* are:

- Polynomial kernel: $K(x_i, x_j) = (x_i \cdot x_j)^d$
- Radial basis function kernel: $K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$ being $\gamma > 0$

3.5.3 K-nearest neighbor

Definition

K-nearest neighbor algorithm (KNN) is a simple algorithm for classifying data points (according to a class label) based on the k closest training points.

KNN is a *non parametric* and *lazy* algorithm, this means that:

- KNN makes no assumption about the data underlying distribution (*non parametric*).
- KNN does not use the *training* points to *generalise* data, actually there is no *training* phase within KNN (*lazy*). However, the *test* phase uses all the *training* data points and this might turn out into a higher cost in terms of time and memory.

Assumptions

KNN makes several assumptions about data itself (not about data underlying distribution, as we have just said):

- Data has to be in a *feature space*, what is more, a notion of *distance* (not necessarily euclidean distance) has to exists within this space.
- A k parameter is given. This parameter controls how many neighbors influence the classification (usually an odd number if the number of classes is 2).

Formulation

Let $X = \{(x_1, \omega_1), \ldots, (x_n, \omega_n)\} : x_i \in \Re^d,\ \omega_i \in \{1, 2, \ldots, \Omega\},\ i = \{1, \ldots, n\}$ be n data samples used as training data.

Let

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + \ldots + (p_d - q_d)^2},\ p, q \in \Re^d$$

be the euclidean distance between samples p and q.

Let $(x, \omega) \notin X : x \in \Re^d,\ \omega \in \{1, 2, \ldots, \Omega\}$ be a *test* sample whose real class ω is unknown.

For $k = 1$, let $\hat{\omega}$ be the predicted class for the test sample x, defined as:

$$\hat{\omega} = \omega_c :\ c = \arg\min_j d(x_j, x),\ \forall j = 1, \ldots, n$$

For $k > 1$, $\hat{\omega}$ will be the most frequent ω among the k nearest training samples to x.

We want to conclude this section by making a couple of comments about KNN regarding with the distance function and the k parameter:

- The formulation we have introduced uses euclidean distance, however, any function that holds the distance function properties can be used.
- The choice of k is critical: a small value will make present noise in data to have a higher influence. On the other hand, a large value will blur the behind philosophy of KNN (points that are near might have the same class) while making KNN computationally more expensive. A maximum value for k would be $k = \sqrt{n}$ while the typical ones are in the range 1, 3, 5, etc.

Chapter 4

Related work

In this chapter the related work in which this research is based is presented.

The research within this master thesis takes as its starting point the work done by Belanche-Muñoz and Blanch back in 2008 titled *Machine learning methods for microbial source tracking* [5], which is described in the next sections.

4.1 Introduction

The paper of Belanche-Muñoz and Blanch [5] reports on a successful application of statistical and inductive learning methods to determine optimal discriminating parameters and develop predictive models for the determination of faecal sources in waters, *recently* and *heavily polluted* with wastewaters.

Thus, the main difference with the work that is being presented in the current thesis is that [5] does not deal with *dilution* nor with *ageing*. All the results that will be introduced within this chapter involve just *recently* and *heavily polluted* water samples.

The main aim is the obtaining of highly accurate predictive models using the lowest number of variables possible.

4.2 Data modelling

In order to find the explanatory models the "Cyprus" dataset has been used, a detailed description can be found in sections 6.1.1 and 6.1.2.

Basically, this dataset is a 103×38 matrix, therefore there are 103 examples and 38 indicators (26 of them are *single* variables while 12 of them are *derived* variables).

81 of the 103 observations (hereafter called the *training set*) were used for the development of predictive models. The remaining 22 observations (the *test set*) presenting unequivocally distinct values according to their origins (11 from human polluted waters and 11 from non-human polluted waters) were hand selected and withheld and will ultimately serve for assessing the goodness of a given model.

In order to estimate the *validation error* or performance of a model leave-one-out cross-validation (LOOCV, see section 3.4.4) was used.

4.3 Experiments and results

4.3.1 Experiment 1: 26 single variables, no feature selection

First experiment carried out consists in building some models using the *training set* with all the 26 *single* variables. Thus, no *feature selection* process is engaged.

Three well-established tools were next used: the Euclidean k-nearest-neighbour technique, KNN (Mitchell T. M. [10]); the linear Bayesian classifier, LBC and the quadratic Bayesian classifier, QBC (Duda et al. [9]). The Euclidean k-nearest- neighbour classifier was analysed for $k = 1$, 2, 3 and 5 (named 1NN, 2NN, 3NN and 5NN, respectively). The two Bayesian classifiers do not need any parameter tuning.

All of the tested methods (KNN, LBC and QBC) provided training set percentages of correct classification higher than 90% when using the 26 single variables (Figure 4.1). QBC provides the highest percentage of correct classification for the training set of observations (98.8%), giving further evidence in favour of a quadratic separation and is the only method achieving 100% test set accuracy.

	TR	TE
1NN	96.3	86.4
2NN	91.4	81.8
3NN	95.1	90.9
5NN	90.1	86.4
LBC	95.1	95.4
QBC	98.8	100

Figure 4.1: Percentages of correct classification provided by the different statistical methods tested for the development of predictive models using the 26 single variables defined in this study. TR: leave-one-out cross-validation performance in the training set (81 observations). TE: performance in the test set (22 observations). (Extracted from [5])

The question remained as to whether smaller models (in terms of number of used variables) can be found while retaining the good performance. All these issues encouraged to make further efforts to select small subsets of variables that keep or increase accuracy and reduce cost.

4.3.2 Experiment 2: all variables, wrapper approach feature selection

Theis experiment was carried out using the whole set of 38 variables (single and derived) and using the Relief algorithm [13] in order to perform feature selection.

According to the list obtained by the Relief algorithm [13], the top three discriminating variables were FRNAPH II, SOMCPH/BTHPH and FC/BTHPH. The first two variables were also clearly distinguished from the rest. The next group in terms of importance was formed by *single* variable FM-FS and *derived* variables FRNAPH II + FRNAPH III and FRNAPH I + FRNAPH IV. The variable HiR was originally included for consideration, but soon discarded after preliminary experiments that consistently ignored it.

The process used to find to find an optimal solution was done as follows:

1. A set S comprising the selected variables was formed, that is S = {BTHPH, SOMCPH, FC, SOMCPH/BTHPH, FC/BTHPH, FRNAPH.II, FM-FS, FRNAPH.II+FRNAPH.III, FRNAPH.I+FRNAPH.IV}.

2. An exhaustive search within S was carried out, using 1NN, LBC, QBC and the SVM as wrappers.

3. Among the resulting subsets that yielded the best LOOCV accuracy in the training set, those sets with the lower number of variables were selected.

4. A second exhaustive search was performed, this time within the full set of 38 variables, but limiting the solutions to those subsets with only two variables, again in a wrapper fashion.

The overall results of this variable selection process were as follows:

Two variables, the ratio SOMCPH/BTHPH and SOMCPH, provided a 100% LOOCV correct classification in the training set for all the learning methods tested (1NN, LBC, QBC and SVM). Additionally, the pair FC/BTHPH and FC also provided excellent results (100%, 98.8%, 98.8% and 100% LOOCV correct classification for 1NN, LBC, QBC and SVM, respectively).

These two combinations of two variables ({SOMCPH/BTHPH, SOMCPH} and {FC/BTHPH, FC}) both gave the right classification for all 22 test observations (100% test accuracy).

4.3.3 Experiment 3: all variables except GA17, wrapper approach feature selection

This experiment was carried out not using the enumeration of phages infecting B. thetaiotaomicron GA17. Consequently, the single variable BTHPH and its derived variables (SOMCPH/BTHPH and FC/BTHPH) were not considered.

This time the procedure was simpler:

Best solutions formed by two and three variables were found by an exhaustive search of the full set of variables (35 of the 38 variables as BTHPH and its derived variables were now removed), again using 1NN, LBC, QBC in a wrapper fashion.

The percentages of correct classification for the most useful methods are shown in figure 4.2. Since this time none of the methods reaches 100%, solutions with four variables were also tested. Note that accuracy is better with increasing numbers of variables. Given the greater demands in computational cost, the SVM method was not used as wrapper to perform feature selection.

Apparently, overall performance was as not as good as in second experiment (see section 4.3.2). Nevertheless, accuracy is well over 95% in the training set and many combinations of methods and subsets of variables still reach 100% in the test set.

4.4 Conclusions

Once the most relevant variables have been selected (consisting of two variables), the distribution plot of observations according to both variables is of great help in

Variables	Size	Cost	Accuracy	100% in test
SOMCPH, FRNAPH II	2	9	96.1 (QBC)	QBC, 1NN
FC, FRNAPH II	2	8	96.1 (QBC)	None
ECP, FRNAPH II	2	13	97.1 (1NN)	QBC, 1NN
BA, FRNAPH II	2	12	97.1 (LBC)	LBC, QBC
BA, FRNAPH II, ECP	3	18	97.1 (LBC)	All
BA, FRNAPH II, FC	3	13	97.1 (LBC)	1NN
FRNAPH I, FRNAPH II, FC	3	15	97.1 (QBC)	All
SOMCPH, FRNAPH II, FRNAPH I	3	16	97.1 (QBC)	LBC, QBC
FRNAPH I, FRNAPH II, ECP	3	20	99.0 (1NN)	QBC, 1NN
ECP, FRNAPH II, BA, SFBIF	4	21	99.0 (QBC)	QBC
SOMCPH, FRNAPH II, BA, SFBIF	4	16	100 (1NN)	QBC

Figure 4.2: Percentages of correct classification provided by the different methods tested for the development of predictive models using the lowest number of variables out of the 38 single and combined variables excluding the single and derived variables that use BTHPH. (Extracted from [5])

determining the criteria for the development of feasible models (figure 4.3).

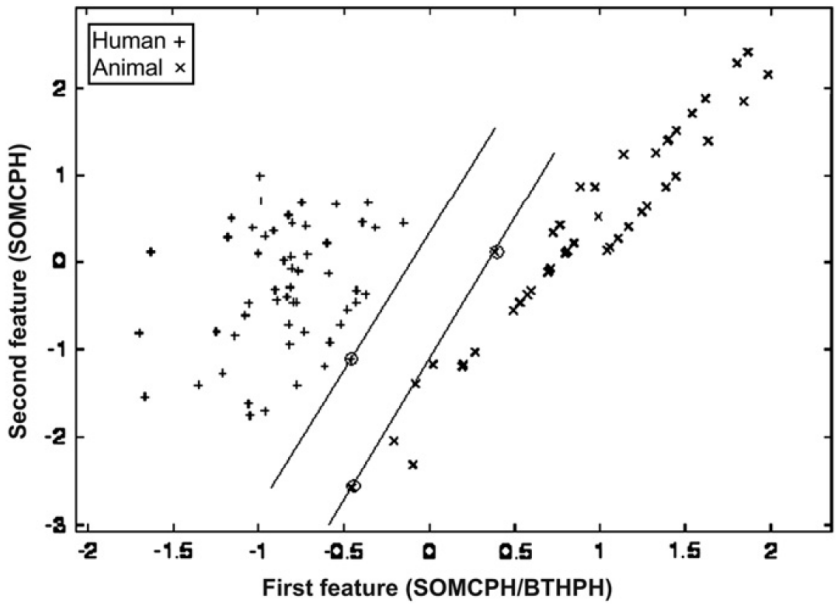

Figure 4.3: Training observations according to the variables SOMCPH/BTHPH and SOMCPH. Variable values are standardized. Superimposed is the maximum-margin linear solution of the SVM. The three support vectors are indicated by a circle. (Extracted from [5])

Looking at the neat bi-dimensional separation and the absence of outliers in-between the two groups of data, the SVM was considered the best choice to obtain a model. This solution is that of maximum distance between the closest known observations from different classes (the margin).

This is, in fact, the starting point of all the research we have developed within this thesis. As have been seen along this section the MST has already been solved to

satisfaction by Belanche-Muñoz and Blanch for water samples that are *recently* and *heavily* polluted (at the *point of source*).

The challenge is now trying to solve the problem for water samples showing different concentration levels and using indicators (variables) with different environmental persistence.

Chapter 5

Methodology

In this section the inner workings of our methodology will be exposed in detail. We will go through all its parts by detailing the challenges that have arisen and how do we manage to solve them.

The main problem we are addressing is designing a methodology which is able to deal and build predictive models not only with *point of source* data (recently and heavily polluted) but also with examples that show different concentration levels (affected by *dilution*) and different environmental persistence (affected by *time*).

In short, the task undertaken is a classification problem consisting in differentiating several fecal pollution origins out of water samples that have not been taken at the *point of source* (they will be referred as *diluted*) and/or present some delay between dumping and measurement time (they will be referred as *aged*). It is very important to mention that both the dilution factor and the aging time are completely unknown when predicting a new example.

5.1 Main strategy

There are many ways of approaching this problem from the machine leaning point of view, in this section some of these strategies to face the problem will be discussed and the main used strategy will be discussed.

Straight solution

A direct and straight solution would consist in simply selecting a set of classifiers, splitting the given data in learning and testing parts, fit the classifiers on the learning part using a resampling technique like cross-validation and choose that combination classifier/parameters that yields the lowest cross-validated error. Assessment of performance could be obtained by evaluating the model on the testing set.

However, this is not going to work. There are at least three technicalities serious enough to hinder or, in the present case, even prevent a standard ML solution:

- The examples in the data matrix are expressed at *point of source*: they were taken right in the spot of contamination. In practice, this will not be the case when predicting a new in the sense that probably it will be *diluted*. Since we want to deal with *dilution* we need predictive models that know how to manage it.

- The examples in the data matrix are expressed at *zero-time*: they were taken right after contamination took place. In practice, this will not happen and the example to be predicted will be *aged*. Moreover, the distinct variables age following different processes that are not completely understood. These processes are altered depending on season conditions. The aging process is merged with the dilution factor and alters even more the relative shape of the two classes, specially considering that each variable evolves differently.

  Given that we want to deal with *aged* examples a way to manage it should be found. Is not enough to have models trained just with *point of source* data.

- An important characteristic of the methodology is that it is based on the data supplied by the user (thus different users can provide different data). Provided data is intended to reflect the information the user has been able to collect about his/her MST study, expressed in the form of specific biological or chemical tracers. This piece of data should be regarded as maximal and a strong need arises to reduce the amount of information needed for the prediction. The immediate consequence is that end-users will supply only a fraction of the variables in the matrix when predicting new examples, depending on varying technical, geographical or monetary conditions.

  From the standpoint of ML, this is a serious concern. Although feature selection can be conducted to reduce the number of variables to a minimum while retaining discriminating ability, the reduced subsets will depend on dilution factor and age. Moreover, there is no guarantee that users will be able to supply all or part of the obtained relevant subsets.

Our strategy: specialized models

Another way of approaching the problem is by building specialized predictive models. By *specialized* we mean models that have been trained with particular data, for instance data that is just *diluted*, just *aged* or at *point of source*.

The strategy consists in building sets of predictive models specialized in *diluted* data. This means they will only be able to make predictions of new *diluted* data samples. Two considerations have to be taken on this:

- Each one of the sets of predictive models will be specialized in one particular dilution degree. A mechanism for estimating the dilution degree of a new example so that it can be predicted by appropriate set of predictive models must be developed (see section 5.2.2).

- Predictive models will not know how to manage *ageing* while new examples to be predicted will be probably *aged*. A mechanism for estimating the age of a new example so that it can be *deage* must be developed (see section 5.2.2).

Next section introduces the inner workings and details of our strategy regarding with the predictive models building.

5.2 Dilution sections approach

The main idea of our strategy, we have called it *dilution sections approach*, is to *recycle* the provided data by using it in a set of independent training processes for different values of the dilution factor, as follows:

Let M_0 be the data provided by the user in matrix form.

Let $V_0 = \{v_1, \ldots, v_m\}$ be the measured variables (indicators) existing in M_0.

1. A set of n non-necessarily equidistant dilution factors $D = \{d_1, \ldots, d_n\}$ is selected on the interval $[1, D_{max}]$.

2. $\forall d_i \in D$ a new diluted data matrix $M_{d_i} = M_0/d_i$ (see section 2.4) is created. This process ends up with a set of data matrices $M = \{M_{d_1}, \ldots, M_{d_i}, \ldots, M_{d_n}\}$.

   Remember from section 2.4.3 that every time a data matrix is diluted is possible that some values will reach their detection threshold and, thus, they will become constant and equal to it.

3. $\forall M_i \in M$ a set of p classifiers $C_i = \{C_{i1}, \ldots, C_{ip}\}$ is created $\forall s \in \mathcal{P}(V_i) : |s| = \{2, 3, 4\}$.

   The reason for choosing all the possible combinations of 2, 3 and 4 variables is, as we stated in section 1, because we are particularly interested in those classifiers that use a minimum number of variables, given the high technical and monetary costs that are implicit in the collection of this kind of data.

The process we have described ends up with n sets of classifiers $C = \{C_1, \ldots, C_i, \ldots, C_n\}$.

Each $C_i \in C$ contains several classifiers trained with data matrix M_i specialized in data whose dilution factor is d_i.

5.2.1 Building and selecting prediction models

There is no restriction when selecting the p classifiers for each C_i. In principle, any classifier can be used. However, we have particular preference and have used these ones:

- Stabilised linear discriminant analysis (see section 3.5.1).
- Support vector machines (see section 3.5.2) using linear, polynomial and radial *kernel* and optimizing the *cost* parameter.
- K-nearest neighbors (see section 3.5.3) optimizing the k parameter.

For each classifier $C_{ip} \in C_i$ we have a measure $E(C_{ip}) \in [0, 1]$ of its *validation error* assessed by 10-fold cross-validation (see section 3.4.3) that allows us to compare models among them in order to choose the best ones.

In order to select the best predictive models (classifiers) for each one of the dilution factors $d_i \in D = \{d_1, \ldots, d_n\}$ we are going to proceed as follows:

$\forall C_i = \{C_{i1}, \ldots, C_{ip}\} \in C$ the $b\%$ best classifiers (b is a methodology parameter) will be selected according to their associated validation error $E(C_{ip})$.

The process of selecting the best models ends up n sets of best classifiers $C' = \{C'_1, \ldots, C'_i, \ldots, C'_n\}$ so that:

- $C'_i = \{C'_{i1}, \ldots, C'_{ib}\} \subset C_i,\ \forall i = \{1, \ldots, n\}$.
- $E(c') \leq E(c),\ \forall c' \in C'_i, c \in C_i, i = \{1, \ldots, n\}$.

The process we have just described implies implicitly a *feature selection* process, what is more, implies a kind of wrapper approach FS process (see section 3.3.2). A number of variables combinations is tested against an *objective function* (in this case a set of p classifiers) and the b best combinations are selected according to the value returned by this *objective function*.

The sets of best classifiers $C' = \{C'_1, \ldots, C'_n\}$ are the ones that are going to make the predictions for new examples. Next sections will explain how in an accurate way.

5.2.2 Dilution and ageing estimation

When a new example has to be predicted the first thing our methodology should do is making an estimation of what is the actual *dilution degree* and *age* of the new example, due to:

- Predictive models are specialized just in *diluted* data. New examples have to be *deaged* before giving them to the models.
- Predictive models are specialized just in a particular *dilution degree*. Therefore, *dilution degree* has to be estimated in order to know which set C'_i of models should predict the example.

Let $V = \{v_1, \ldots, v_n\}$ the actual values of the n variables of the new example. Those variables are potentially *diluted* and *aged*.

Let $\tilde{V} = \{\tilde{v_1}, \ldots, \tilde{v_n}\}$ the unknown non-diluted zero-time values of the n variables of the new example.

Consider $\{V_1, \ldots, V_n\}$ the names of the variables of the new example.

Consider $S_i = \{(x_{i1}, \log_{10}(y_{i1})), \ldots, (x_{im}, \log_{10}(y_{im}))\}$ a univariate data sample that contains m measurements that represent how variable V_i persists in time. Let $f_i(x) = ax + b$ represent the regression line on S_i.

Consider $S_{i\alpha} = \{(x_{i1}, \log_{10}(\frac{y_{i1}}{\alpha})), \ldots, (x_{im}, \log_{10}(\frac{y_{im}}{\alpha}))\}$ the S_i data sample *diluted* by a factor of α. Let $f_{i\alpha}(x) = ax + b - \log_{10}(\alpha)$ represent the regression line on $S_{i\alpha}$.

Consider now a variable V_i belonging to the new example with known measured value v_i, the theoretical equation for variable V_i is $v_i = \log_{10}(\frac{\tilde{v_i}}{\alpha}) + a_i t$ being a_i the obtained slope and $\tilde{v_i}$ the unknown non-diluted zero-time measurement.

In prediction time, we are given a collection of available variables $\{V_1, \ldots, V_n\}$ with measured values $\{v_1, \ldots, v_n\}$. As we have seen, the theoretical equations for these variables need the value of α (which is completely unknown). However, if we subtract one pair (i, j) of these theoretical equations we arrive at:

$$(a_i - a_j)t + \log_{10}(\tilde{v_i}) - \log_{10}(\tilde{v_j}) = v_i - v_j$$

This means the difference in time behaviour of a pair of variables is not affected by the dilution process (at least not until one of the variables reaches its detection threshold).

The quantities $\log_{10}(\tilde{v_i})$ and $\log_{10}(\tilde{v_j})$ within the above equation are not directly available but, recalling that the supplied data matrix M_0 consists of non-diluted zero-time measurements, a system of $\frac{n(n-1)rows(M_0)}{2}$ equations can be obtained by replacing

every pair $\log_{10}(\tilde{v_i})$ and $\log_{10}(\tilde{v_j})$ from the last equation with the corresponding values in the data matrix.

Having all that, the estimation t^* of the *age* of the example is done as follows:

Algorithm 2 Estimating age

eq_system $\leftarrow$ *nil*
for $i = 1 \rightarrow n - 1$ **do**
  for $j = i + 1 \rightarrow n$ **do**
    for $k = 1 \rightarrow rows(M_0)$ **do**
      $(a_i - a_j)t = v_i - v_j - \log_{10}(M_0[k, V_i]) + \log_{10}(M_0[k, V_j])$
      equation is added to *eq_system* (only t is unknown).
    end for
  end for
end for
$t^* \leftarrow$ Solve *eq_system* by OLS method
return t^*

Once the *age* is estimated the estimation α^* of the *dilution degree* is done as follows:

Algorithm 3 Estimating dilution degree

eq_system $\leftarrow$ *nil*
for $i = 1 \rightarrow n$ **do**
  $\log_{10}(\alpha) = t^* a_i + b_i - v_i$ equation is added to *eq_system* (only α is unknown).
end for
$\alpha^* \leftarrow$ Solve *eq_system* by OLS method
return 10^{α^*}

At this point both the *age* and the *dilution degree* of the new example have been estimated the fecal origin of the new example can be predicted. Next section introduces how is this prediction done in a detailed way.

5.2.3 New examples prediction

At this point we have:

- A new example $V = \{v_1, \ldots, v_p\}$ whose fecal pollution origin has to be predicted.
- A set of variables labels $\{V_1, \ldots, V_p\}$ of the new example.
- An estimation t^* of its *time persistence* (*age*).
- An estimation α^* of its *dilution degree*.

Remember that all the best classifiers $C'_{ij} : i = \{1, \ldots, n\}$ are specialized in predicting *non-aged* examples that have a dilution factor of d_i. Since the new example to be predicted is potentially *aged* and *diluted* it has to be *deaged* for it to be just *diluted*. This is done in this way:

$$\forall v_i \in V : i = \{1, \ldots, p\},\ \hat{v}_i = v_i - a_i t^* \text{ where:}$$

- $\hat{v}_i$ is the *deaged* value of the (still diluted) variable V_i.
- a_i is the slope of the regression line representing the *time persistence* of variable V_i.

At this moment we have a *deaged* (but still diluted) example $\hat{V} = \{\hat{v}_1, \ldots, \hat{v}_p\}$. Now the fecal origin of the example $\hat{V}$ can be done in this way:

Consider C'_i the set of best classifiers that are specialized in predicting examples that have a dilution factor of $d_i \in D = \{d_1, \ldots, d_n\}$.

Consider $V(c) : |V(c)| \in \{2, 3, 4\}$ the set of variables that classifier c uses to make its prediction.

Consider $P(c)$ the prediction done by the classifier c.

Prediction is done by the classifiers belonging to $C'_i : i = \{1, \ldots, n\}$ classifiers set where d_i is the nearest value of set $D = \{d_1, \ldots, d_n\}$ to α^*.

Once the particular set of classifiers C'_i is set, the actual set of predictions is defined as:

$$Pr = \{pr_1, \ldots, pr_q\} = \{P(c)\ \forall c \in C'_i : V(c) \subseteq \{V_1, \ldots, V_p\}\}.$$

Majority class within set Pr will be the final predictions.

In case a tie is produced the final prediction will be the *majority class* according to data provided by the user in matrix form M_0.

5.3 Validation

Our methodology also provides a frame for assessing the quality of the predictive models created so far.

The main idea is, once the predictive models are built and the best of them are select, generating a representative *validation set* and then predict each one of these examples as explained in section 5.2.3. We understand as a representative *validation set* the one that:

- Has a large enough number of examples in order to obtain a fair validation assessment.
- Has examples that are both *diluted* and *aged*.

But how can this *validation set* be built? Remember we are only provided with a matrix M_0 with a finite number of examples that are measured at the *point of source*. However, since we know how *dilution* and *ageing* can be numerically modelled, we could build a *validation set* as big as we desire in the way that algorithm 4 shows.

Algorithm 4 returns a *validation set* with *n_examples* examples. Note that:

- $\frac{M_{r,i}}{d} + a_i t$ is just *diluting* and *ageing* the provided *point of source* value, in this case $M_{r,i}$.
- a_i is the slope of the regression line representing the *time persistence* of the i-th variable.
- Each example belonging to the *validation set* preserves the class of the original example $M_{r,*}$ from which it was generated.

Algorithm 4 Building validation set

Input: $M \leftarrow m \times n$ matrix with user input data
Input: $D_{max} \leftarrow$ maximum dilution degree factor
Input: $T_{max} \leftarrow$ maximum ageing hours
Input: $n\_examples \leftarrow$ desired number of examples for the validation set
  $validation\_set \leftarrow$ list()
  for $i = 1 \rightarrow n\_examples$ **do**
    $r \leftarrow$ pick a random number from the interval [1,m] uniformly
    $d \leftarrow$ pick a random number from the interval [1,D_{max}] uniformly
    $t \leftarrow$ pick a random number from the interval [1,T_{max}] uniformly
    $example_{1\times n} \leftarrow \forall i : i = \{1, \ldots, n\},\ \frac{M_{r,i}}{d} + a_i t$
    add $example_{1\times n}$ to $validation\_set$
  end for
  return $validation\_set$

Once *validation set* is built the only step that remains to assess the validation of the methodology is predicting each one of the examples belonging to the *validation set* as explained in section 5.2.3.

The actual methodology performance will be the number of well classified examples from *validation set* divided by its total number of examples.

Chapter 6

Application

This chapter is focused on testing the implementation we have done of our methodology (described in chapter 5) with two different real data examples.

Both data examples consist of:

- A *maximal* matrix with all the collected data, it is intended to have all the *examples* and *indicators* of the whole MST study. It will be used mainly for *training* and building the predictive models.
- Some matrices with test data, it is not intended to have all the *examples* and *indicators* of the whole MST study. This matrix should contain only feasible examples that any MST practitioner would provide, this means these examples will not contain as many *indicators* as the first matrix. It will be used mainly for *testing* how good are the models built so far.
- User available information on how do the indicators persist in time. This information can be both a regression line or several measurements of the indicators along the time.

For both real data examples we will describe basically: how it has been collected (its origin), what is the feedback provided by the implementation after analysing the *training* matrix and, finally, predicting the *testing* data and showing the results.

6.1 Real test dataset 1: the "Cyprus" data

6.1.1 Data origin

The "Cyprus" data belongs to the European project TOFPSW (Tracking the origin of fecal pollution in surface water) [14]. According to [14], TOFPSW focused on tracking the origin of fecal pollution in surface waters by the use of standardised or established methods.

Within TOFPSW a certain amount of data was collected from different European geographic areas and, finally, it was embodied into a matrix form that we called the "Cyprus" matrix.

"Cyprus" matrix has been provided to us by professor Anicet R. Blanch, from Microbiology Department at Barcelona University.

6.1.2 Data description

The MST study that lead to the so-called "Cyprus" matrix was originally designed to focus on four key components:

1. The study focused only on the differentiation between *human* and *non-human* sources.

2. Included data is highly-polluted data because failures previously reported in the literature were often related to the use of diluted samples.

3. Included data was collected around widely different geographical areas.

4. Several indicators of fecal pollution from both *human* and *non-human* sources throughout the study, since this is needed for defining ratios between discriminant and non-discriminant indicators and for defining the persistence of the values of fecal contaminants in the environment.

Let us focus in where does data come from, as we have just said data was collected from different geographical areas (see figure 6.1). Concretely, a multi-laboratory study was undertaken in the following areas of Europe: northern Europe (Stockholm, Sweden), northwestern Europe (Brighton, United Kingdom), central Europe (Nancy, northeastern France), southeastern Europe (Nicosia, Cyprus), and southwestern Europe (Barcelona, Spain).

Source	Geographic area									
	Spain		France		Sweden		United Kingdom		Cyprus	
	No. of samples	No. of sampling sites	No. of samples	No. of sampling sites	No. of samples	No. of sampling sites	No. of samples	No. of sampling sites	No. of samples	No. of sampling sites
Human wastewater										
Urban	22	6	10	1	18	9	22	3	5	1
Hospital			16	1	5	5				
Military camp									17	1
Animal wastewater[a]										
Cow	6	2	15	3	9	6	8	1		
Pig	9	3			5	1	7	1	8	1
Poultry	8	2			4	1	7	1	9	1
Horse					4	3				
Mixed[b]	4	2	7	2					5	1
Total	49	15	48	7	45	25	44	6	44	5

[a] Slaughterhouses or farm slurries.
[b] Cow, pig, and sheep.

Figure 6.1: Distribution of samples of "Cyprus" matrix according to the geographical areas. (Extracted from [3]).

Figure 6.1 shows the number of samples and sampling sites for each one of the geographical areas. Apart from that the origin of the wastewater is also reflected: for instance, human samples come from urban, hospitals and military camps while animal samples origin are cow, pig, poultry, horse and a mixture of them.

Figure 6.2 shows all the measured variables (indicators) that will form the *training* matrix. As can be seen two kind of variables exists: on one hand there are *single* variables while on the other hand there are *derived* variables, which are the result of some sort of combination of two *single* variables.

Variable	Label	Parameter
Single	BA	Detection of the presence (1) or absence (0) of *Bifidobacterium adolescentis*
	BE	Detection of the presence (1) or absence (0) of *Bifidobacterium dentium*
	BTHPH	Enumeration of *B. fragilis* bacteriophages using the new host strain *B. thetaiotaomicron* GA17
	CHOL	Concn of cholestanol or 5-α-colestan-3ß-ol
	CL	Enumeration of clostridia
	COP	Concn of coprostanol or 5ß-cholestan-3ß-ol
	EPICOP	Concn of epicoprostanol or 5ß-cholestan-3α-ol
	ETHYLCOP	Concn of stigmastanol or 24-ethylcoprostanol
	FC	Enumeration of fecal coliforms
	FE	Enumeration of fecal enterococci
	FRNAPH	Enumeration of F-specific RNA bacteriophages
	FRNAPH I	% of genotype I of F-specific RNA bacteriophages
	FRNAPH II	% of genotype II of F-specific RNA bacteriophages
	FRNAPH III	% of genotype III of F-specific RNA bacteriophages
	FRNAPH IV	% of genotype IV of F-specific RNA bacteriophages
	FTOTAL	Enumeration of F-specific bacteriophages
	RYC2056	Enumeration of *B. fragilis* bacteriophages using the host strain RYC2056
	SFBIF	Enumeration of sorbitol-fermenting bifidobacteria
	SOMCPH	Enumeration of somatic coliphages
	TBIF	Enumeration of total bifidobacteria
Derived	CNFC	% of cellobiose-negative fecal coliforms
	COP/EPICOP	Ratio of concn of coprostanol to that of epicoprostanol
	COP/ETHYLCOP	Ratio of concn of coprostanol to that of stigmastanol
	DA	Sum of values of BA and BE
	DiC	Simpson's diversity index for fecal coliforms
	DiE	Simpson's diversity index for enterococci
	ECP	% of *E. coli* Ph-Plate phenotypes
	FC/BTHPH	Ratio of the no. of fecal coliforms to that of the new host strain *B. thetaiotaomicron* GA17
	FC/FE	Ratio of the no. of fecal coliforms to that of enterococci
	FC/RYC2056	Ratio of the no. of fecal coliforms to that of phages on the host strain *B. fragilis* RYC2056
	FC/SOMCPH	Ratio of the no. of fecal coliforms to that of coliphages
	FMFS	% of *E. faecium* and *E. faecalis*
	FRNAPH I + FRNAPH IV	Sum of the % of genotypes I and IV of F-specific RNA bacteriophages
	FRNAPH II + FRNAPH III	Sum of the % of genotypes II and III of F-specific RNA bacteriophages
	HiR	% of *E. hirae*
	SFBIF/TBIF	Ratio of the no. of sorbitol-fermenting bifidobacteria to that of total bifidobacteria
	SOMCPH/BTHPH	Ratio of the no. of somatic coliphages to that of the new host strain *B. thetaiotaomicron* GA17
	SOMCPH/RYC2056	Ratio of the no. of somatic coliphages to that of phages on the host strain *B. fragilis* RYC2056

Figure 6.2: Indicators measured for "Cyprus" matrix.(Extracted from [3]).

Initially, the group of variables taken into consideration was the 20 *single* variables plus a group of 6 *derived* variables from the phenotyping of fecal coliforms and enterococci: percentage of cellobiose-negative fecal coliforms (CNFC), diversity index for fecal coliforms (DiC), diversity index for enterococci (DiE), percentage of E. coli Ph-Plate phenotypes (ECP), FMFS, and HiR.

However, in order to improve the chances of obtaining good predictive models, 12 new variables were derived by combining some of the 20 single variables (as sums or ratios).

Samples containing incomplete or outlying parameter values likely to be attributable to technical error were discarded. Consequently, a training data matrix with 38 variables and 103 samples was finally obtained.

6.1.3 Importance of variables regarding with prediction

First step of our methodology is building and selecting the best prediction models using the "Cyprus" matrix we have been provided with. Detail explanation of this step can be found at section 5.2.1.

In order to provide the user information about what are the best variables for predicting new examples the methodology implementation returns figure 6.3: the X-axis corresponds to the indicators while the Y-axis corresponds to all the dilution sections, from 1 to 61, being the lower ones the ones with fewer dilution factor. Given

a variable v and a dilution section d, the reddish (darker in B/W paper) the cell is, the more important is variable v predicting examples for the section d dilution factor.

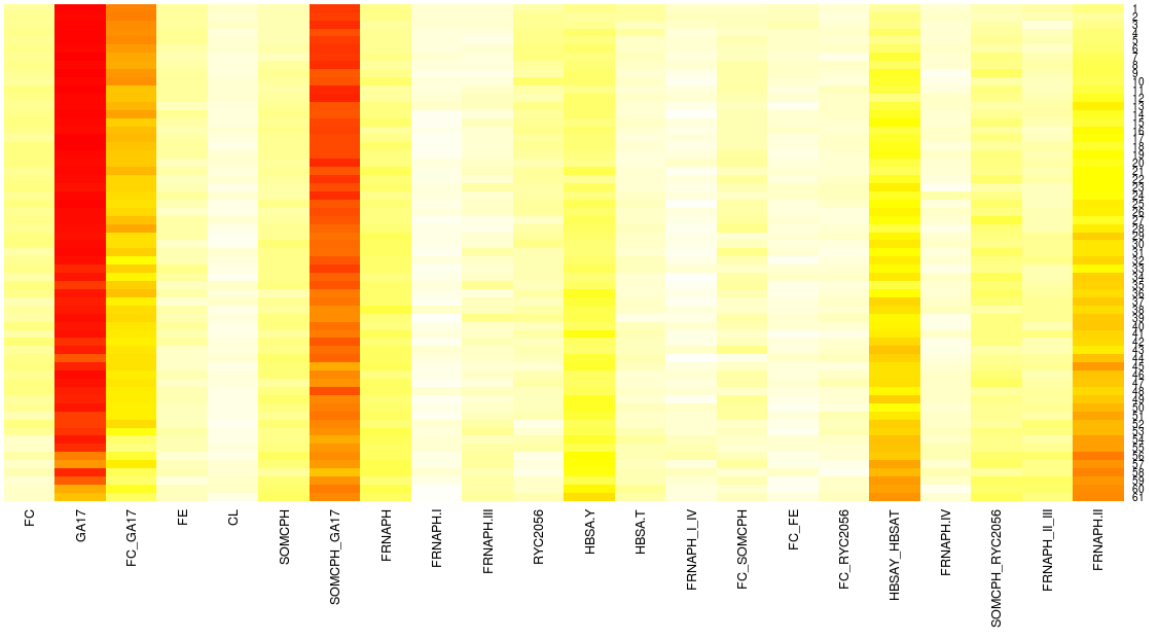

Figure 6.3: Importance of variables from "Cyprus" matrix regarding with prediction.

This information is very important for the user because it allows to know whether a particular variable has good or bad discrimination power. Usually, measuring a variable is an expensive process. Thus, knowing that a particular variable is not useful can save a high amount of money, because it will not be measured again.

Looking at figure 6.3 we can observe several things:

- In global terms, *GA17* and the ratio *SOMCPH/GA17* are the indicators with more discrimination power along all the dilution sections.
- At very low dilution factors, the ratio *FC/GA17* has also good discrimination power.
- At very high dilution factors, *FRNAPH.II* and the ratio $HBSA.Y/HBSA.T$ are also good indicators.

The rest of the indicators have not much discrimination power when comparing them with the indications that have been mentioned so far. This does not mean these indicators and their combinations are no able to make accurate predictions, probably they do; however, the above mentioned indicators are, by far, much better than the others on the prediction task. User knows now that making an effort to provide *GA17*, *SOMCPH*, *FC*, *FRNAPH.II*, $HBSA.Y$ and $HBSA.T$ indicators on new examples will lead the system to make predictions with higher accuracy.

Let us illustrate the difference between good and bad variables by plotting these variables values for all the 103 examples of the training matrix. We will show good and bad variables under low dilution conditions as well as good variables under high dilution conditions.

Good indicators with no dilution

Figures 6.4 and 6.5 show two examples of good combinations of variables under no dilution conditions.

In particular, figure 6.4 shows two variables: the ratio formed by *SOMCPH* and *GA17* and the ratio formed by *FC* and *GA17*. On the other hand, figure 6.5 shows *GA17* and the ratio formed by *SOMCPH* and *GA17*.

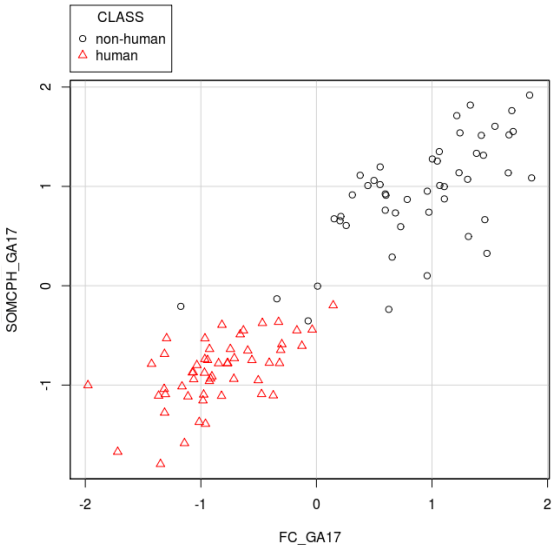

Figure 6.4: SOMCPH/GA17 and FC/GA17 indicators.

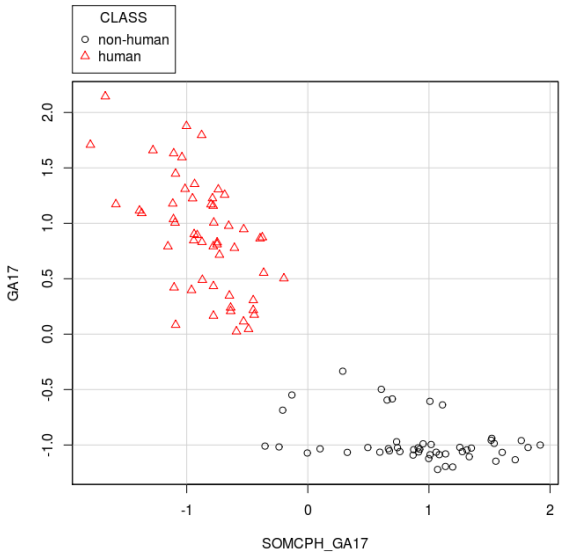

Figure 6.5: GA17 and SOMCPH/GA17 indicators.

Both figures show good combinations of variables since in both a separation among the classes can be established. However, if we look carefully at both figures, in the second one (figure 6.5) classes are intuitively easier to separate than figure 6.4 since this one has a kind of conflicting zone where both classes meet, while on figure 6.5 a linear separation can be made with enough margin.

This is an expected result since in figure 6.3 we can see that, for no dilution conditions, *GA17* and *SOMCPH/GA17* indicators are reddish (darker in B/W paper) and therefore more important than *FC/GA17*.

Bad indicators with no dilution

Figures 6.6 and 6.7 show two examples of bad combinations of variables under no dilution conditions.

In particular, figure 6.6 shows two variables: *FRNAPH.I* and *CL*. On the other hand, figure 6.7 shows *FRNAPH.III* and the *FE*.

Both combinations of variables show a poor separability since there exists a high degree of overlap between the two classes in both cases.

This is an expected result since in figure 6.3 we can see that, for low dilution conditions, *FRNAPH.I*, *CL*, *FRNAPH.III* and *FE* indicators are almost in white colour for no dilution; this means that they are not important, that is, that they have very poor discriminating power.

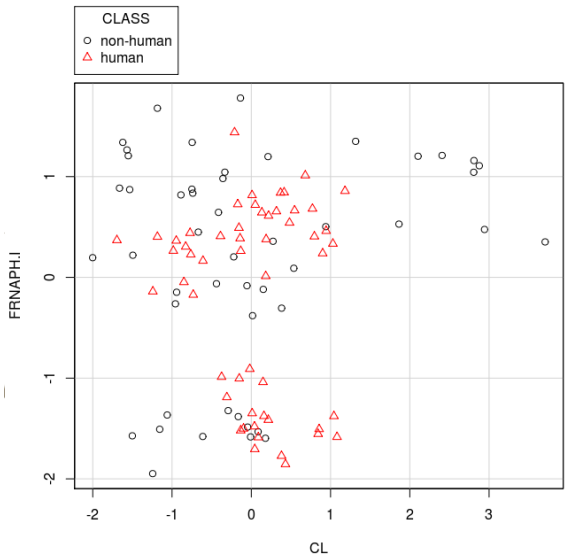

Figure 6.6: FRNAPH.I and CL indicators.

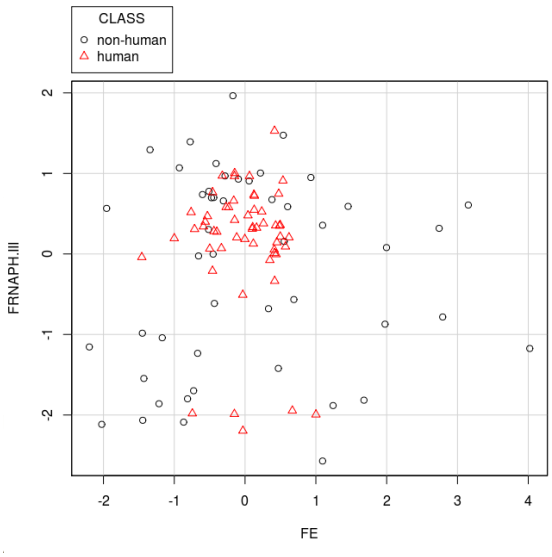

Figure 6.7: FRNAPH.III and FE inds.

Good indicators with high dilution

Figures 6.4 and 6.5 show two examples of good combinations of variables under high dilution conditions. More precisely, variable have been diluted by a dilution factor of 1000.

In particular, figure 6.8 shows two variables: *GA17* and the ratio formed by *SOMCPH* and *GA17*. On the other hand, figure 6.9 shows the ratio formed by *HBSA.Y* and *HBSA.T* along with *FRNAPH.II* indicator.

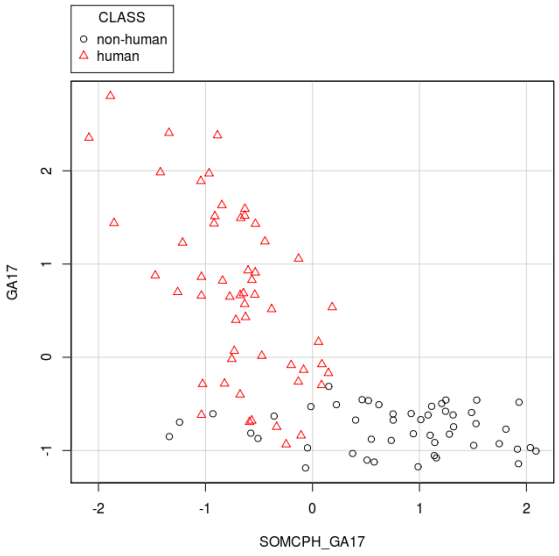

Figure 6.8: GA17 and SOMCPH/GA17 indicators.

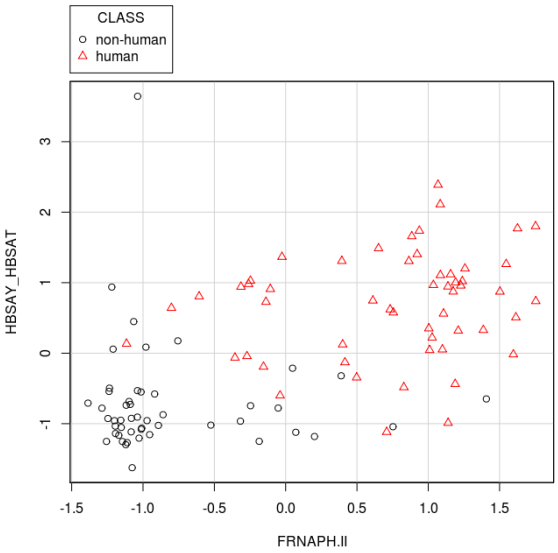

Figure 6.9: HBSA.Y/HBSA.T and FRNAPH.II indicators.

Both figures show good combinations of variables since in both a reasonably good separation among the classes can be established, being this separation easier in figure

Table 6.1: Importance of variables from "Cyprus" matrix regarding with dilution and age estimation.

Attribute set	NRMSE
GA17 , HBSA.Y	0.48
CL , HBSA.Y , HBSA.T	0.51
FRNAPH.I , HBSA.T	0.57
CL , HBSA.Y	0.58
FRNAPH.I , FRNAPH.IV , HBSA.T	0.61
GA17 , HBSA.T	0.61
SOMCPH , HBSA.Y , HBSA.T	0.64
CL , HBSA.T	0.65
FE , FRNAPH.I , HBSA.T	0.66
SOMCPH , HBSA.T	0.67

6.8 than in figure 6.9.

At this point is specially interesting to compare figures 6.4 and 6.5 (good variable under low dilution conditions) with figures 6.8 and 6.9. If we look carefully all these figures we can see that when dilution is added the problem (separating both classes) has become harder even using good variables, as expected (see section 2.5.2 about dilution effect).

6.1.4 Importance of variables regarding with time and age estimation

Besides assessing the importance of the variables regarding with prediction the methodology implementation also delivers information about how good or bad are the variables when estimating the age and the dilution (see section 5.2.2 on how this estimation is done).

This particular given feedback are the best indicators sets and their *normalised root mean square error* (NRMSE) when estimating age and dilution (as shown in table 6.1).

In the particular case of the "Cyrpus" matrix best indications for estimating age and dilution are *GA17*, *HBSA.Y*, *CL*, *HBSA.Y*, *HBSA.T*. As well as with the indicators that are good on prediction, user has to decide if it worths to make an effort to provide these indicators, since better and more confident predictions will be done if the are present in the new examples to be predicted.

6.1.5 Validation

Validation of the models created using the "Cyprus" matrix has been done as explained in section 5.3.

The accuracy of the validation process using 1000 generated samples is **80.81%** of well-classified examples.

Nevertheless, in order to establish an upper-bound estimation of the accuracy another validation was done, this time using the real *dilution degree* and *age* of the example instead of estimating them. The result of this new validation is **90.56%**.

The first (and fair) validation process is just an estimation of what is the accuracy of the methodology over the "Cyrpus" matrix. In the next section real data will be used and we will see whether this estimation is pessimistic or optimistic.

6.1.6 Test matrix prediction

Within this section two "Cyprus" test data matrices will be introduced. All their examples will be predicted (see section 5.2.3) using the best build models (see section 5.2.1).

6.1.6.1 Diluted and aged data

First "Cyprus" test data matrix we are introducing is the more challenging of both. It contains 50 *environmental* examples (this means they are prone to be *diluted* and *aged*) with at most 5 indicators per example.

In fact, the large majority of examples contain just 3 indicators (*FC*, *SOMCPH* and *GA17*), while only a few of examples contain *RYC2056*, *HBSA.T* and *HBSA.Y*. It is known that all the examples within this test matrix have human pollution origin.

Each one of the examples in matrix has been predicted as explained in section 5.2.3 and these are the results:

- 44 out of 50 examples were predicted as *human* origin pollution.
- The accuracy of our methodology on this test matrix is **88%**.

6.1.6.2 Point of source data

Second "Cyprus" test data matrix we are introducing is not as challenging as the one introduced in last section. It contains 15 *point of source* examples (this means they are nor diluted nor aged) with at least 10 indicators per example.

Each one of the examples in matrix has been predicted as explained in section 5.2.3 and these are the results:

- The pollution origin of 14 out of 15 examples was predicted correctly.
- The accuracy of our methodology on this test matrix is **93.33%**.

6.1.6.3 Conclusions

The results obtained on both "Cyprus" test data matrices are promising.

The actual accuracy within the first matrix (section 6.1.6.1) should be considered very promising for several reasons:

1. The data in this matrix is challenging. It is composed by *environmental* data, which means that it is supposed to be quite *diluted* and *aged*. It contains at most 5 indicators per example.

2. Despite data difficulty, a particularly high accuracy of **88%** is achieved.

3. Dr. Anicet R. Blanch has assured that misclassified examples are particularly difficult for the pollution origin to be established.

4. Confidence of the system in most of the misclassified examples is low (around 60%), which means that system is not so sure about the prediction it has done.

On the other hand, second "Cyprus" test matrix (section 6.1.6.2) is a lot easier than the one we have just talked about. It is composed by 15 *fresh* examples (which means they are not *diluted*, just taken at the *point of source*).

The results we have obtained with this test matrix are also good, for two reasons:

1. We have misclassified just one of the 15 examples, this means we have an accuracy of 93.33%

2. Dr. Anicet R. Blanch has assured again that the single misclassified example is not very clear for the pollution origin to be established. The degree of confidence of the system while predicting this particular example is about 60%, which means that it is not sure at all about the prediction.

6.2 Real test dataset 2: the "Delta" data

6.2.1 Data origin

"Delta" data was collected at Ebre river delta, located in the south of Catalonia, north-eastern Spain. Location is interesting since it offers a low and fuzzy pollution scenario, in this sense it was a challenge for the indicators and methodologies to be tested.

Another interesting fact is that the region council informed that in that place some controversy exists in order to know who is responsible for the pollution detected in the beaches nearby, whether it was a water treatment plant or a poultry farm.

Figure 6.10: "Delta" data measurement place.

Figure 6.11: "Delta" data measurement place.

Figures 6.10 and 6.11 show two different places at Ebre river delta where measurement were done.

"Delta" data has been provided to us by professor Anicet R. Blanch.

6.2.2 Data description

The "Delta" matrix is composed by 20 examples and 12 indicators . The examples do not have just *human* or *non-human* origin; instead of it in "Delta" matrix examples can have *human*, *cow*, *poultry* and *pig*.

"Delta" matrix is, in principle, more difficult than "Cyprus" matrix due to two reasons:

1. It has just 20 examples, against the 103 examples of the "Cyprus" matrix.
2. Examples can have 4 different classes, not just 2 as in the "Cyprus" matrix.

The combination of having just 20 examples and 4 classes to discriminate makes the problem, a priori, quite challenging.

The 12 indicators from "Delta" matrix are divided into two groups: 7 of them are called *cultivated* while the other 5 are considered *molecular*.

6.2.3 Importance of variables regarding with prediction

First step of our methodology is building and selecting the best prediction models using the "Delta" matrix we have been provided with. Detail explanation of this step can be found at section 5.2.1.

The strategy we have followed with "Delta" matrix is slightly different from what was used within "Cyprus" matrix, main differences are:

1. Predictive models are now trained with both *diluted* and *aged* data.
2. When predicting new examples there will be no estimation of the dilution degree or the age. Instead of it, all the best models of all the sections will participate in the prediction.

The study of the importance of the indicators regarding with prediction will have several parts. First of all the indicators will be divided into three groups:

1. Using only *cultivated* indicators.
2. Using only *molecular* indicators.
3. Using all the indicators.

For each one of the groups the indicators will be analysed for summer and winter since, given one particular indicators, its time persistence behaves different depending on the season.

Next figures will show the importance of the indicators in this way: the X-axis corresponds to the indicators while the Y-axis corresponds to all the dilution and ageing sections, from 1 to 30, being the lower ones the ones with fewer dilution factor and age. Given a variable v and a section d, the reddish the cell is, the more important is variable v when predicting examples for the section d dilution factor and age.

Cultivated indicators

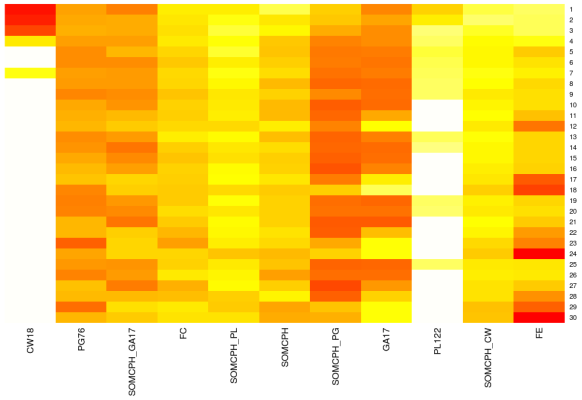

Figure 6.12: Importance of variables: cultivated indicators in summer.

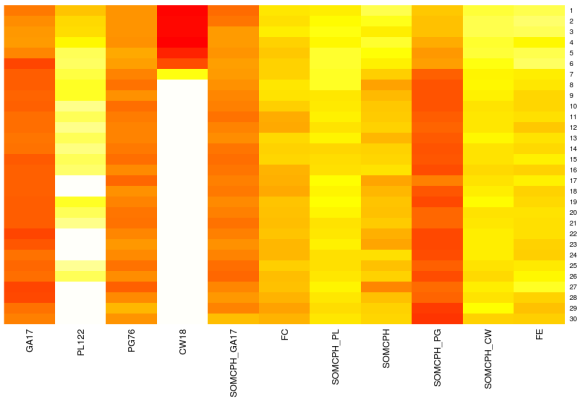

Figure 6.13: Importance of variables: cultivated indicators in winter.

Figure 6.12 shows that within a low *dilution* degree and low *age* scenario the best indicators are: CW18, SOMCPH/GA17, GA17, PG76 and PL122. As long as *dilution* degree and *age* grow CW18 is no more useful while PG76, SOMCPH/GA17 and GA17 still are good indicators. At high time persistence (*age*) FE indicator is clearly the best of them all.

However, on figure 6.13 we can see that most of the indicators (except CW18) that are good within a low *dilution* degree and low *age* scenario are also good when *dilution* degree and *age* grow, as expected. The reason is that on winter all the indicators are less affected by the *time persistence* (they remain on water during more time).

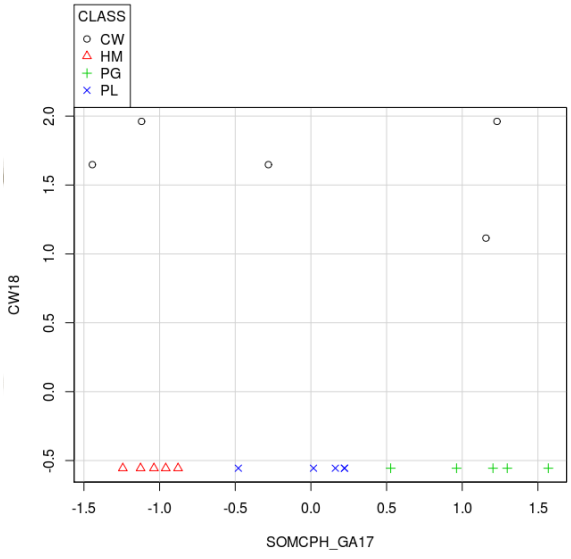

Figure 6.14: Good cultivated indicators: CW18 and SOMCPH/GA17.

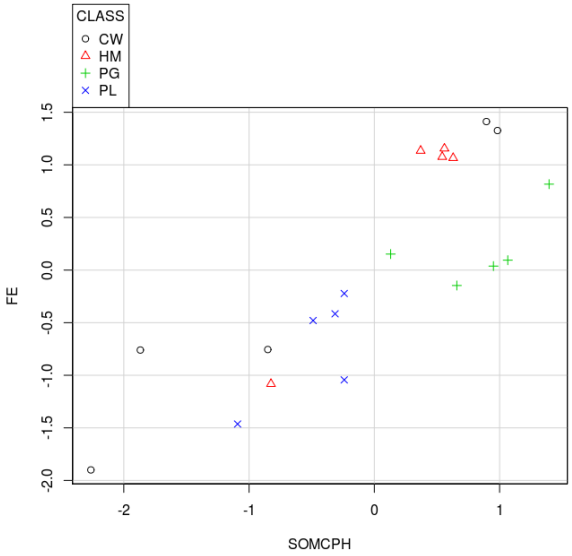

Figure 6.15: Bad cultivated indicators: FE and SOMPCH.

On figures 6.14 and 6.15 we can see two examples of good and bad cultivated indicators, respectively.

On the left figure we can see that the combination of CW18 and SOMCPH/GA17 is able to separate all four classes perfectly. Therefore, a model using these indicators will be able to generalize correctly the data.

On the other hand, right figure shows the combination of FE and SOMCPH. While some separation of the classes can still be done the chosen variables do not allow to generalize data as good as on the other figure.

Plots from figures 6.14 and 6.15 are consistent with the information shown in figures 6.12 and 6.13. According to these figures CW18 and SOMCPH/GA17 are more important than FE and SOMCPH. Figure 6.14 shows that, as expected, CW18 and SOMCPH/GA17 indicators allow to built better models than the other two variables.

Molecular indicators

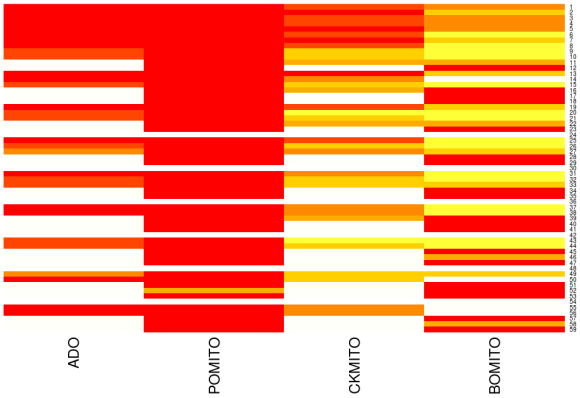

Figure 6.16: Importance of variables: molecular indicators in summer.

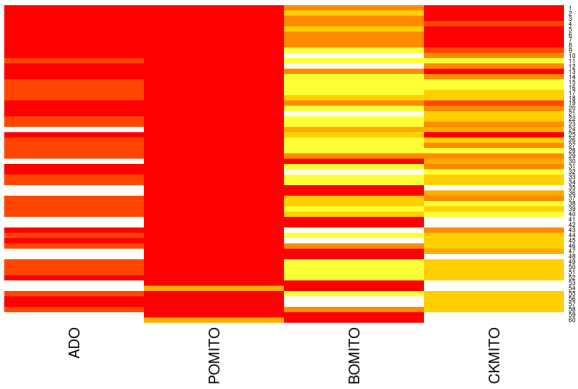

Figure 6.17: Importance of variables: molecular indicators in winter.

Figure 6.16 shows that the overall best indicators in summer season are ADO and POMITO. Notice that CKMITO is good at low *dilution* degree and low *age* while BOMITO is better at high *time persistence*.

In the same way as with the *cultivated* indicators, on figure 6.17 we can see that in winter indicators persist more in time, being the best ones ADO and POMITO again. As expected (in winter all the indicators are less affected by the *time persistence*) BOMITO is again better at high *time persistence* while CKMITO persists better in time.

Important thing to be noticed that DEN indicator does not even appear on no one of both figures. This means it has much less importance that the other four indicators.

On figures 6.18 and 6.19 we can see two examples of good and bad molecular indicators, respectively.

On the left figure we can see that the combination of ADO, POMITO and CKMITO is able to separate all four classes perfectly. Therefore, a model using these indicators will be able to generalize correctly the data. Notice that some of the corners in the cube have more than one examples superposed.

On the other hand, right figure shows the combination of DEN and BOMITO. While some separation of the classes can still be done the chosen variables do not allow to generalize data as good as on the other figure. Notice that at the (0,0)

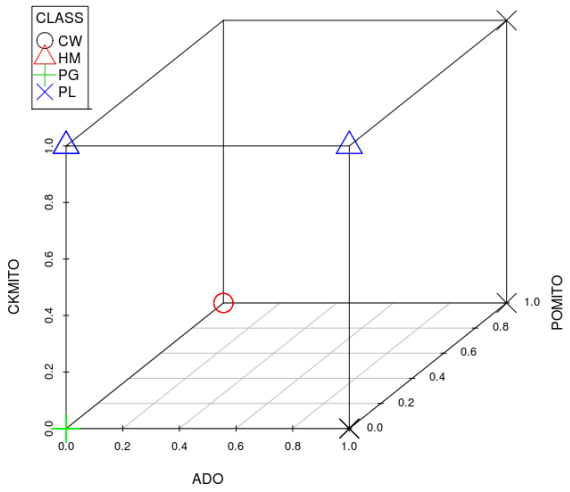

Figure 6.18: Good molecular indicators: ADO, POMITO and CKMITO

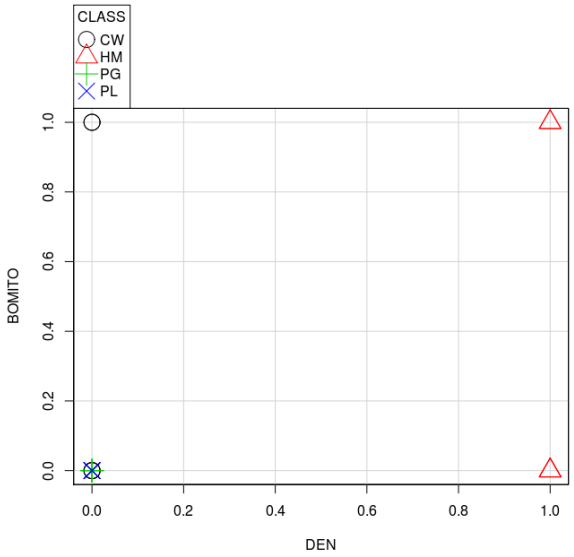

Figure 6.19: Bad molecular indicators: DEN and BOMITO

point there are superposed examples of three different classes, which are impossible to discriminate using only these two indicators.

Again, plots from figures 6.18 and 6.19 are consistent with the information shown in figures 6.16 and 6.17. According to these figures ADO, POMITO and CKMITO are more important than DEN and BOMITO. Figure 6.18 shows that, as expected, ADO, POMITO and CKMITO indicators allow to built better models than the other two ones.

All indicators

When mixing both *cultivated* and *molecular* indicators things are kind of similar as before (see figure 6.20). As we may guessed, within a low *dilution* degree and low *age* scenario best indicators are CW18, PG76, GA17, SOMCPH/GA17, BOMITO, ADO and CKMITO. As long as *dilution* degree and *age* grow the indicators start losing importance (CW18 disappearing faster than the others) while some others like FE seem to be particularly good at high *time persistence* levels.

Figure 6.21 shows the importance of all indicators in winter season. As expected, most of the indicators persist more in time than in summer; nevertheless, good indicators in summer are, in general terms, good indicators in winter.

6.2.4 Validation

Validation of the models created using the "Delta" matrix has been done as explained in section 5.3.

We have divided the validation process into three groups according to the variables that are used, for each of the groups always 1000 generated samples have been used.

The accuracy of the validation process using just *cultivated* variables is **86.2%** of well-classified examples.

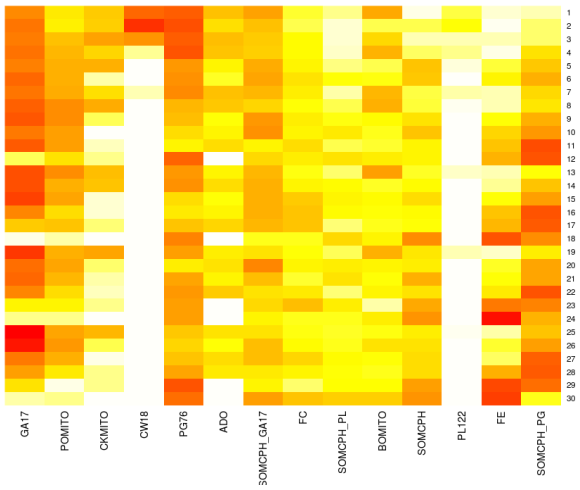

Figure 6.20: Importance of variables: all indicators in summer.

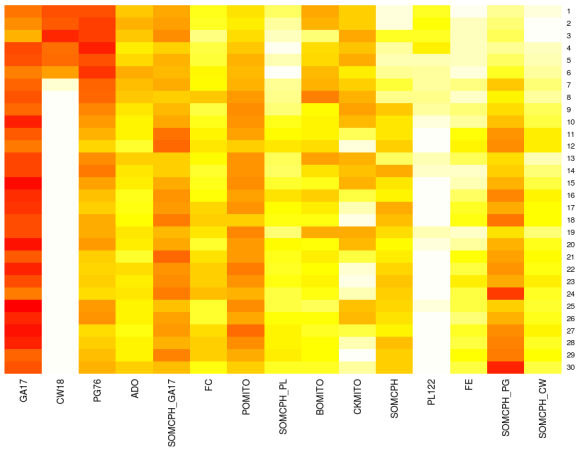

Figure 6.21: Importance of variables: all indicators in winter.

The accuracy of the validation process using just *molecular* variables is **87.93%** of well-classified examples.

The accuracy of the validation process using all variables is **88.6%** of well-classified examples.

The validation process is just an estimation of what is the accuracy of the methodology over the "Delta" matrix. In the next section real data will be used and we will see whether this estimation is pessimistic or optimistic.

6.2.5 Test matrix prediction

Within this section two "Delta" test data matrices will be introduced. As we have said before, each one of the examples will be predicted using the best of models of all the sections.

6.2.5.1 "Llobregat" test matrix

"Llobregat" test matrix is composed by 11 examples, all of them have human origin and are potentially *diluted* and *aged*.

Results (shown in table 6.2) obtained when predicting these examples can be considered very good since:

- 10 out of 11 examples (**91%**) were classified correctly.
- Dr. Anicet R. Blanch has assured that the wrong classified example was a little bit confusing for it to be predicted. Moreover, system confidence when predicting this particular examples was quite low.

Considering those items the results of this test can definitely be considered very good results.

Table 6.2: Results of "Llobregat" test matrix.

Predicted class	Prediction confidence
Pig	43.6%
Human	71.6%
Human	83.6%
Human	73.0%
Human	80.4%
Human	97.5%
Human	85.5%
Human	73.1%
Human	68.5%
Human	69.7%
Human	76.8%
Human	67.8%
Human	65.2%
Human	73.3%
Human	76.3%

6.2.5.2 "Ebre" test matrix

The "Ebre" test matrix is composed by 32 examples, all of them have mainly poultry origin and are potentially *diluted* and *aged*.

However, the zone in which this measurements were done is near a populated area and it is known that the are also some cows in the zone. In this sense, possible traces of human and cow originated fecal pollution could be found.

This test matrix is considered very challenging since the values of the indicators are very low, this means they are very *diluted* and/or *aged*. In fact, pollution is so low in this water that it is actually not harmful for humans.

Results (shown in table 6.3) obtained when predicting these examples can be considered very good since:

- Most of the predicted classes are poultry and this is known to be the main fecal pollution origin in the zone.
- A few examples were predicted as cow and human origin. Moreover, the only example predicted as human was measured near a water treatment plant.

Considering those items we consider that the results are very satisfactory since it reflects very well the adequate proportion of different fecal pollution origins that exist in the zone. Results has also been assessed by Dr. Anicet R. Blanch.

Table 6.3: Results of "Ebre" test matrix.

Predicted class	Prediction confidence
Poultry	55.2%
Poultry	39.5%
Cow	42.8%
Poultry	49.2%
Poultry	47.7%
Cow	42.9%
Cow	41.3%
Poultry	69.9%
Cow	38.7%
Cow	42.0%
Poultry	43.2%
Human	59.4%
Cow	37.7%
Cow	39.6%
Poultry	40.9%
Poultry	56.4%
Poultry	44.0%
Poultry	34.9%
Poultry	37.6%
Poultry	73.2%
Poultry	50.2%
Poultry	42.4%
Poultry	51.7%
Poultry	73.0%
Poultry	52.3%
Cow	36.5%
Poultry	52.5%
Poultry	90.9%
Poultry	52.9%
Poultry	41.5%
Poultry	46.8%
Poultry	74.1%

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this document we have widely discussed the research done so far within the context of the master thesis belonging to the *Artificial Intelligence Master Program* organized by *Universitat Politècnica de Catalunya (UPC)*, *Universitat de Barcelona (UB)* and *Universitat Rovira i Virgili (URV)*.

The problem was to design a methodology which is able of solving the MST problem dealing with *diluted* and *aged* data.

Before introducing the actual methodology some MST concepts as well as all the learning methods and techniques used along this research have been exposed. A review of the related work, which is the base of this master thesis, has also been discussed in depth. Finally, the theoretical methodology as well as its application on two real examples have been introduced.

The results emerged from this application can be considered very promising since real data was used and they have been assessed by an expert. These results will be the basis for keeping working in order to improve our methodology.

7.2 Future work: the no-dilution paradigm

In this section we will discuss the future work related to our research, that will be actually done within the next months. It involves defining a new problem paradigm in which we can get rid of the dilution effects.

7.2.1 Motivation

During this paper we have exposed how did we manage with both *dilution* and *ageing* processes. Remember we wanted to create a methodology which is able to deal with *diluted* and *aged* data.

However, during recent discussions we realized that we could get rid of dilution by making the next assumptions:

1. As exposed in section 2.4.1, the dilution (or *concentration level*) just depends on the volume of water in which the measurement is done. More volume implies less *concentration level*.

2. Despite water volume is increased the actual indicator particles will still be there, they do not disappear.

3. An extremely precise measuring device that was able to measure any indicator level within any water volume would remove the dilution effect over the measurement.

We have supposed precise measurements can be done; for this reason, here in advance *dilution* will have no more sense and we will only going to concentrate in dealing with *ageing*.

The fact of getting rid of dilution makes the problem easier. Remember from section 2.5.2 that once a water sample is *aged* or *diluted* the problem becomes harder since both classes start to blend one over another.

Therefore, dropping one of the two process will reduce this blending and separating both classes will not be as hard as before.

7.2.2 Strategy

Predictive models building

The strategy for the predictive models to be build is analogue as the one exposed in section 5.2. The aim is building specialized predictive models that are just trained with *aged* data.

Formulation will be exactly as in section 5.2 but, instead of having dilution factors $d_i \in D = \{d_1, \ldots, d_n\}$ we will have time sections $t_i \in T = \{t_1, \ldots, t_n\}$.

Therefore, $\forall t_i \in T$ a new *aged* data matrix M_{t_i} (see section 2.4) is created. This process ends up with that a set of data matrices $M = \{M_{t_1}, \ldots, M_{t_i}, \ldots, M_{t_n}\}$. The rest of predictive models building process is done in the same way as in section 5.2.1.

Predicting a new example

Several discussions lead us to two strategies in order to predict a new (potentially *aged* and *diluted*) example, further research and experimentation will make us decide by one or another:

1. First strategy is analogue to the one described on section 5.2.3. It involves estimating the *dilution factor* and the *age* of a new example. Dilution factor means nothing and estimated *age* is now used to decide which section of the best models will participate in the prediction.

2. Second strategy is easier: there is no *age* estimation on the new example since all the sections of the best models will participate in the prediction.

We want to conclude this section by clarifying that, despite new examples will be potentially *diluted* and *aged* and models will trained only with *aged* data, our new paradigm is still feasible.

The reason lies in the fact that all the training data is *standardised* before training the models. Given that dilution only implies a division of all indicators of a sample (see section 2.4.1) the actual *standardised* sample will be the same no matter if sample is *diluted* or not.

References

[1] Hagedorn, C., Blanch, A.R., Harwood, V.J. Microbial Source Tracking: Methods, Applications and Case Studies New York: Springer, 2011. ISBN 978-1-4419-9385-4.

[2] Malakoff, D. Microbiologists in the trail of polluting bacteria. In *Science*, 295:2352-2353.

[3] Blanch, A.R., Belanche-Muñoz, L., Bonjoch, X., Ebdon, J., Gantzer, C., Lucena, F., Ottoson, J., Kourtis, C., Iversen, A., Kuhn, I., Moce, L., Muniesa, M., Schwartzbrod, J., Skraber, S., Papageorgiou, G.T., Taylor, H., Wallis, J., Jofre, J. Integrated analysis of established and new microbial and chemical methods for microbial source tracking. In *Appl. Environ. Microbiol.* 72 (9), 5915-5926, 2006.

[4] Bonjoch, X., Lucena, F., Blanch, A.R. The persistence of bifidobacteria populations in a river measured by molecular and culture techniques In *Journal of Applied Microbiology*, 107: pp. 1178-1185, 2009.

[5] Belanche, L., Blanch, A.R. Machine learning methods for microbial source tracking. In *Environmental Modelling & Software*, 23(6): pp. 741-750, 2008.

[6] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

[7] Cherkassky, V., Mulier, F. Learning from data: concepts, theory and methods. Second Edition. New Jersey: Wiley-Interscience, 2007. ISBN 978-0-471-68182-3.

[8] Russell, S., Norvig, P. Artificial intelligence: a modern approach. New Jersey: Prentice Hall, 2003. ISBN 0-13-080302-2.

[9] Duda, R.O., Hart, P.E., Stork, L. Pattern Classification. New York: John Wiley & Sons, 2003. ISBN 978-0471056690.

[10] Mitchell, M. Machine Learning. New York: McGraw-Hill Higher Education, 1997. ISBN 978-0070428072.

[11] Vapnik, V. Statistical Learning Theory. New Jersey: John Wiley & Sons, 1998. ISBN 978-0471030034

[12] Burges, C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition In *Data Mining and Knowledge Discovery*, 2: pp. 121-167, 1998.

[13] Kira, K., Rendell, L.A. The feature selection problem: traditional methods and a new algorithm. In *Proceedings of the Ninth National Conference on Artificial Intelligence* AAAI Press, pp. 129-134, 1992.

[14] TOFPSW Tracking the origin of faecal pollution in surface water. http://www.eugris.info/DisplayProject.asp?P=4311

[15] Vapnik, V. The Nature of Statistical Learning Theory. New York: Springer, 1995. ISBN 978-0387987804

[16] Webb, A. Statistical Pattern Recognition. New York: John Wiley & Sons, 2002. ISBN 0-470-84514-7.

[17] Peters A., Hothorn T. Ipred: improved predictors URL http://cran.r-project.org/web/packages/ipred/ipred.pdf p. 45

Outcomes

A part of this research has been presented at 2nd Workshop on Applications of Pattern Analysis, celebrated in Octorber 2011 in Castro Urdiales (Spain).

The reference for this paper is:

Sanchez, D., Belanche L.A., Blanch A.R. A Software System for the Microbial Source Tracking Problem. In *JMLR Workshop and Conference Proceedings*, 17: pp. 56-62, 2011.